

Cut Elimination for a Logic with Generic Judgments and Induction

Alwen Tiu

Computer Sciences Laboratory
The Australian National University

Abstract. This paper presents a cut-elimination proof for the logic LG^ω , which is an extension of a proof system for encoding generic judgments, the logic $FO\lambda^{\Delta\nabla}$ of Miller and Tiu, with an induction principle. The logic LG^ω , just as $FO\lambda^{\Delta\nabla}$, features extensions of first-order intuitionistic logic with fixed points and a “generic quantifier”, ∇ , which is used to reason about the dynamics of bindings in object systems encoded in the logic. A previous attempt to extend $FO\lambda^{\Delta\nabla}$ with an induction principle has been unsuccessful in modeling some behaviours of bindings in inductive specifications. It turns out that this problem can be solved by relaxing some restrictions on ∇ , in particular by adding the axiom $B \equiv \nabla x.B$, where x is not free in B . We show that by adopting the equivariance principle, the presentation of the extended logic can be much simplified. This paper contains the technical proofs for the results stated in [14]; readers are encouraged to consult [14] for motivations and examples for LG^ω .

1 Introduction

This work aims at providing a framework for reasoning about specifications of deductive systems using *higher-order abstract syntax* [10]. Higher-order abstract syntax is a declarative approach to encoding syntax with bindings using Church’s simply typed λ -calculus. The main idea is to support the notions of α -equivalence and substitutions in the object syntax by operations in λ -calculus, in particular α -conversion and β -reduction. There are at least two approaches to higher-order abstract syntax. The *functional programming* approach encodes the object syntax as a data type, where the binding constructs in the object language are mapped to functions in the functional language. In this approach, terms in the object language become values of their corresponding types in the functional language. The *proof search* approach encodes object syntax as expressions in a logic whose terms are simply typed, and functions that act on the object terms are defined via relations, i.e., logic programs. There is a subtle difference between this approach and the former; in the proof search approach, the simple types are inhabited by well-formed expressions, instead of values as in the functional approach (i.e., the abstraction type is inhabited by functions). The proof search approach is often referred to as *λ -tree syntax* [7], to distinguish it from the functional approach. This paper concerns the λ -tree syntax approach.

Specifications which use λ -tree syntax are often formalized using hypothetical and generic judgments in intuitionistic logic. It is enough to restrict to the fragment of first-order intuitionistic logic whose only formulas are those of hereditary Harrop formulas, which we will refer to as the *HH* logic. Consider for instance the problem of defining the data type for untyped λ -terms. One first introduces the following constants:

$$app : tm \rightarrow tm \rightarrow tm \quad abs : (tm \rightarrow tm) \rightarrow tm$$

where the type tm denotes the syntactic category of λ -terms and app and abs encode application and abstraction, respectively. The property of being a λ -term is then defined via the following theory:

$$\bigwedge M \bigwedge N (lam\ M \wedge lam\ N \Rightarrow lam\ (app\ M\ N)) \ \& \\ \bigwedge M ((\bigwedge x. lam\ x \Rightarrow lam\ (M\ x)) \Rightarrow lam\ (abs\ M))$$

where \bigwedge is the universal quantifier and \Rightarrow is implication.

Reasoning about object systems encoded in HH is reduced to reasoning about the structure of proofs in HH . McDowell and Miller formalize this kind of reasoning in the logic $FO\lambda^{\Delta\mathbb{N}}$ [3], which is an extension of first-order intuitionistic logic with fixed points and natural numbers induction. This is done by encoding the sequent calculus of HH inside $FO\lambda^{\Delta\mathbb{N}}$ and prove properties about it. We refer to HH as object logic and $FO\lambda^{\Delta\mathbb{N}}$ as meta logic. McDowell and Miller considered different styles of encodings and concluded that explicit representations of hypotheses and, more importantly, eigenvariables of the object logic are required in order to capture some statements about object logic provability in the meta logic [4]. One typical example involves the use of hypothetical and generic reasoning as follows: Suppose that the following formula is provable in HH .

$$\bigwedge x.pxs \Rightarrow \bigwedge y.pyt \Rightarrow pxt.$$

By inspection on the inference rules of HH , one observes that this is only possible if s and t are syntactically equal. This observation comes from the fact that the right introduction rule for universal quantifier, reading the rule bottom-up, introduces new constants, or eigenvariables. The quantified variables x and y will be replaced by distinct eigenvariables and hence the only matching hypothesis for pxt would be pxs , and therefore s and t has to be equal. Let $\vdash_{HH} F$ denote the provability of the formula F in HH . Then in the meta logic, we would want to be able to prove the statement:

$$\forall s \forall t. (\vdash_{HH} \bigwedge x.pxs \Rightarrow \bigwedge y.pyt \Rightarrow pxt) \supset s = t.$$

The question is then how we would interpret the object logic eigenvariables in the meta logic. It is demonstrated in [4] that the existing quantifiers in $FO\lambda^{\Delta\mathbb{N}}$ cannot be used to capture the behaviours of object logic eigenvariables directly. McDowell and Miller then resort to a non-logical encoding technique (in the sense that no logical connectives are used) which has some similar flavor to the use of deBruijn indices. The use of this encoding technique, however, has a consequence that substitutions in the object logic has to be formalized explicitly.

Motivated by the above mentioned limitation of $FO\lambda^{\Delta\mathbb{N}}$, Miller and Tiu later introduce a new quantifier ∇ to $FO\lambda^{\Delta\mathbb{N}}$ which allows one to move the binders from the object logic to the meta logic. A generic judgment in the object logic, for instance $\vdash_{HH} \bigwedge x.Gx$ is reflected in the meta logic as $\nabla x. \vdash_{HH} Gx$. This meta logic, called $FO\lambda^{\Delta\nabla}$ [8], allows one to perform case analyses on the provability of the object logic. Tiu later extended $FO\lambda^{\Delta\nabla}$ with induction and co-induction rules, resulting in the logic Linc [13]. However, some inductive properties about the object logic are not provable in Linc. For example, the fact that $\vdash_{HH} \bigwedge x.Gx$ implies $\forall t. \vdash_{HH} Gt$ (that is, the extensional property of universal quantification) is not provable in Linc. As it is shown in [13], this is partly caused by the fact that $B \equiv \nabla x.B$, where x is not free in B , is not provable in Linc or $FO\lambda^{\Delta\nabla}$. In this paper we present the logic LG^ω , which is an extension of $FO\lambda^{\Delta\nabla}$ with natural number induction and with the axiom schemes:

$$\nabla x \nabla y. Bxy \supset \nabla y \nabla x. Bxy \quad \text{and} \quad B \equiv \nabla x.B \tag{1}$$

where x is not free in B in the second scheme. We show that inductive properties of λ -tree syntax specifications can be stated directly and in a purely logical fashion, and proved in LG^ω .

Relation to nominal logic In formulating the proof system for LG^ω , it turns out that we can simplify the presentation a lot if we adopt the idea of *equivariant predicates* from nominal logic [11]. That is, provability of a predicate is invariant under permutations of *names*. This is technically done by introducing a countably infinite set of name constants into the logic, and change the identity rule of the logic to allow equivalence under permutations of name constants:

$$\frac{\pi.B = \pi'.B'}{\Gamma, B \multimap B'} id$$

where π and π' are permutations on names. LG^ω is in fact very close to nominal logic, when we consider only the behaviours of logical connectives. In particular, the quantifier ∇ in LG^ω shares the same properties, in relation to other connectives of the logic, with the \forall quantifier in nominal logic. However, there are two

important differences in our approach. First, we do not attempt to redefine α -conversion and substitutions in LG^ω in terms of permutations (or *swapping*) and the notion of *freshness* as in nominal logic. Name swapping and freshness constraints are not part of the syntax of LG^ω . These notions are present only in the meta theory of the logic. In LG^ω , for example, variables are always considered to have empty support, that is, $\pi.x = x$ for every permutation π . This is because we restrict substitutions to the “closed” ones, in the sense that no name constants can appear in the substitutions. A restricted form of open substitutions can be recovered indirectly at the meta theory of LG^ω . The fact that variables have empty support allows one to work with permutation free formulas and terms. So in LG^ω , we can prove that $p\ x\ a \supset p\ x\ b$, where a and b are names, without using explicit axioms of permutations and freshness. In nominal logic, one would prove this by using the swapping axiom $p\ x\ a \supset p\ ((a\ b).x)\ ((a\ b).b)$, where $(a\ b)$ denotes a swapping of a and b , and then show that $(a\ b).x = x$. The latter might not be valid if x is substituted by a , for example. The validity of this formula in nominal logic would therefore depend on the assumption on the support of x .

The second difference between LG^ω and nominal logic is that LG^ω allows closed terms (again, in the sense that no name constants appear in them) of type *tm*, while in nominal logic, allowing such terms would lead to an inconsistent theory in nominal logic [11]. As an example, the type *tm* in the encoding of λ -terms mentioned previously can be treated as a nominal type in LG^ω . This has an important consequence that we do not need to redefine the notion of substitutions for the encoded λ -terms. For example, we can define the (lazy) evaluation relation on untyped λ -terms as the theory:

$$\begin{aligned} eval\ (abs\ M)\ (abs\ M) &\equiv \top \\ eval\ (app\ M\ N)\ V &\equiv eval\ M\ (abs\ P) \wedge eval\ (P\ N)\ V \end{aligned}$$

without having to explicitly define substitutions on terms of type *tm* inside LG^ω . Substitutions in the object language in this case is modelled by β -reduction in the meta-language of LG^ω .

Outline of the paper Section 2 introduces the logic LG , which is an extension of first order intuitionistic logic with a notion of name permutation and the ∇ -quantifier. LG serves as the core logic for a more expressive logic, LG^ω , which is obtained by adding rules for fixed points, equality and induction to LG . Section 3 examines several properties of derivations, in particular, those that concern preservation of provability under several operations on sequents, e.g., substitutions. Section 4 defines the cut reduction, used in the cut-elimination proof. The cut elimination proof itself is an adaptation of the cut-elimination proof of $FO\lambda^{\Delta\nabla}$ by McDowell and Miller [3], which makes use of the reducibility technique. Section 5 defines the normalizability and the reducibility relations which are crucial to the cut elimination proof in Section 6. Finally, in Section 7, we show that the proof system LG is actually equivalent to $FO\lambda^{\Delta\nabla}$ (without fixed points and equality) with non-logical rules corresponding to the axioms given in (1) above.

This paper contains the technical proofs for the results stated in [14]; readers are encouraged to consult [14] for motivations and examples for LG^ω .

2 A logic for generic judgments

We first define the core fragment of the logic LG^ω which does not have fixed point rules or induction. The starting point is the logic $FO\lambda^\nabla$ introduced in [8]. $FO\lambda^\nabla$ is an extension of a subset of Church’s Simple Theory of Types in which formulas are given the type o . The core fragment of LG^ω , which we refer to as LG , shares the same set of connectives as $FO\lambda^\nabla$, namely, \perp , \top , \wedge , \vee , \supset , \forall_τ , \exists_τ and ∇_τ . The type τ in the quantifiers is restricted to that which does not contain the type o . Hence the logic is essentially first-order. We abbreviate $(B \supset C) \wedge (C \supset B)$ as $B \equiv C$.

The sequents of $FO\lambda^{\Delta\nabla}$ are expressions of the form

$$\Sigma; \sigma_1 \triangleright B_1, \dots, \sigma_n \triangleright B_n \multimap \sigma_0 \triangleright B_0$$

where Σ is a signature, i.e., a set of eigenvariables scoped over the sequent and σ_i is a local signature, i.e., list of variables locally scoped over B_i . The introduction rules for ∇ , reading the rules bottom-up, introduce new

local variables to the local signatures, just as the right introduction rule of \forall introduces new eigenvariables to the signature. The expression $\sigma_i \triangleright B_i$ is called a local judgment, and is identified up to renaming of variables in σ_i . This enforces a limited notion of equivariance: for example $a \triangleright pa \multimap b \triangleright pb$ is provable, since both local judgments are equivalent up to renaming of local signatures. However, the judgments $(a, c) \triangleright pa$ and $b \triangleright pb$ are considered distinct judgments, and so are $(a, b) \triangleright qab$ and $(b, a) \triangleright qab$. These restrictions are relaxed in LG .

The sequent presentation of LG can be simplified, that is, without using the local signatures, if we employ the equivariance principle. For this purpose, we introduce a distinguished set of base types, called *nominal types*, which is denoted with \mathcal{N} . Nominal types are ranged over by ι . We restrict the ∇ quantifier to nominal types. For each nominal type $\iota \in \mathcal{N}$, we assume an infinite number of constants of that type. These constants are called *nominal constants*. We denote the family of nominal constants by $\mathcal{C}_{\mathcal{N}}$. The role of the nominal constants is to enforce the notion of equivariance: provability of formulas is invariant under permutations of nominal constants. Depending on the application, we might also assume a set of non-nominal constants, which is denoted by \mathcal{K} .

We assume the usual notion of capture-avoiding substitutions. Substitutions are ranged over by θ and ρ . Application of substitutions is written in a postfix notation, e.g., $t\theta$ is an application of θ to the term t . Given two substitutions θ and θ' , we denote their composition by $\theta \circ \theta'$ which is defined as $t(\theta \circ \theta') = (t\theta)\theta'$. A *signature* is a set of variables. A substitution θ respects a given signature Σ if there exists a set of typed variables Σ' such that for every $x : \tau$ in the domain of θ , it holds that $\mathcal{K} \cup \Sigma' \vdash \theta(x) : \tau$. We denote by $\Sigma\theta$ the minimal set of variables satisfying the above condition. We assume that variables, free or bound, are of a different syntactic category from constants.

Definition 1. A permutation on $\mathcal{C}_{\mathcal{N}}$ is a bijection from $\mathcal{C}_{\mathcal{N}}$ to $\mathcal{C}_{\mathcal{N}}$. The permutations on $\mathcal{C}_{\mathcal{N}}$ are ranged over by π . Application of a permutation π to a nominal constant a is denoted with $\pi(a)$. We shall be concerned only with permutations which respect types, i.e., for every $a : \iota$, $\pi(a) : \iota$. Further, we shall also restrict to permutations which are finite, that is, the set $\{a \mid \pi(a) \neq a\}$ is finite. Application of a permutation to an arbitrary term (or formula), written $\pi.t$, is defined as follows:

$$\begin{aligned} \pi.a &= \pi(a), \text{ if } a \in \mathcal{C}_{\mathcal{N}}. & \pi.c &= c, \text{ if } c \notin \mathcal{C}_{\mathcal{N}}. & \pi.x &= x \\ \pi.(M \ N) &= (\pi.M) (\pi.N) & \pi.(\lambda x.M) &= \lambda x.(\pi.M) \end{aligned}$$

A permutation involving only two nominal constants is called *swapping*. We use $(a \ b)$, where a and b are constants of the same type, to denote the swapping $\{a \mapsto b, b \mapsto a\}$.

The *support* of a term (or formula) t , written $\text{supp}(t)$, is the set of nominal constants appearing in it. It is clear from the above definition that if $\text{supp}(t)$ is empty, then $\pi.t = t$ for all π . The definition of Σ -substitution implies that for every θ and for every $x \in \text{dom}(\theta)$, $\theta(x)$ has empty support. Therefore Σ -substitutions and permutations commute, that is, $(\pi.t)\theta = \pi.(t\theta)$.

A sequent in LG^ω is an expression of the form $\Sigma; \Gamma \multimap C$ where Σ is a signature. The free variables of Γ and C are among the variables in Σ . The inference rules for the core fragment of LG^ω , i.e., the logic LG , is given in Figure 1. In the rules, the typing judgment $\Sigma, \mathcal{K}, \mathcal{C}_{\mathcal{N}} \vdash t : \tau$ denotes the typability of $t : \tau$, given the typing context $\Sigma \cup \mathcal{K} \cup \mathcal{C}_{\mathcal{N}}$ in Church's simple type system.

In the $\nabla\mathcal{L}$ and $\nabla\mathcal{R}$ rules, a denotes a nominal constant. In the $\exists\mathcal{L}$ and $\forall\mathcal{R}$ rules, we use *raising* [6] to encode the dependency of the quantified variable on the support of B , since we do not allow Σ -substitutions to mention any nominal constants. In the rules, the variable h has its type raised in the following way: suppose \vec{c} is the list $c_1 : \iota_1, \dots, c_n : \iota_n$ and the quantified variable x is of type τ . Then the variable h is of type: $\iota_1 \rightarrow \iota_2 \rightarrow \dots \rightarrow \iota_n \rightarrow \tau$. This raising technique is similar to that of $FO\lambda^{\Delta\nabla}$, and is used to encode explicitly the minimal support of the quantified variable. Its use prevents one from mixing the scopes of \forall (dually, \exists) and ∇ . That is, it prevents the formula $\forall x \nabla y. pxy \equiv \nabla y \forall x. pxy$, and its dual, to be proved.

Looking at the introduction rules for \forall and \exists , one might notice the asymmetry between the left and the right introduction rules. The left rule for \forall allows instantiations with terms containing any nominal constants while the raised variable in the right introduction rule of \forall takes into account only those which are in the support of the quantified formula. However, we will see that we can extend the dependency of the

raised variable to an arbitrary number of fresh nominal constants not in the support without affecting the provability of the sequent (see Lemma 17 and Lemma 18).

$$\begin{array}{c}
\frac{\pi.B = \pi'.B'}{\Sigma; \Gamma, B \vdash B'} id_{\pi} \quad \frac{\Sigma; \Delta_1 \vdash B_1 \quad \dots \quad \Sigma; \Delta_n \vdash B_n \quad \Sigma; B_1, \dots, B_n, \Gamma \vdash C}{\Sigma; \Delta_1, \dots, \Delta_n, \Gamma \vdash C} mc \quad \frac{\Sigma; \Gamma, B, B \vdash C}{\Sigma; \Gamma, B \vdash C} c\mathcal{L} \\
\\
\frac{}{\Sigma; \Gamma, \perp \vdash C} \perp\mathcal{L} \quad \frac{}{\Sigma; \Gamma \vdash \top} \top\mathcal{R} \\
\\
\frac{\Sigma; \Gamma, B_i \vdash C}{\Sigma; \Gamma, B_1 \wedge B_2 \vdash C} \wedge\mathcal{L}, i \in \{1, 2\} \quad \frac{\Sigma; \Gamma \vdash B \quad \Sigma; \Gamma \vdash C}{\Sigma; \Gamma \vdash B \wedge C} \wedge\mathcal{R} \\
\\
\frac{\Sigma; \Gamma, B \vdash C \quad \Sigma; \Gamma, D \vdash C}{\Sigma; \Gamma, B \vee D \vdash C} \vee\mathcal{L} \quad \frac{\Sigma; \Gamma \vdash B_i}{\Sigma; \Gamma \vdash B_1 \vee B_2} \vee\mathcal{R}, i \in \{1, 2\} \\
\\
\frac{\Sigma; \Gamma \vdash B \quad \Sigma; \Gamma, D \vdash C}{\Sigma; \Gamma, B \supset D \vdash C} \supset\mathcal{L} \quad \frac{\Sigma; \Gamma, B \vdash C}{\Sigma; \Gamma \vdash B \supset C} \supset\mathcal{R} \\
\\
\frac{\Sigma, \mathcal{K}, \mathcal{C}_{\mathcal{N}} \vdash t : \tau \quad \Sigma; \Gamma, B[t/x] \vdash C}{\Sigma; \Gamma, \forall x.B \vdash C} \forall\mathcal{L} \quad \frac{\Sigma, h; \Gamma \vdash B[h\vec{c}/x]}{\Sigma; \Gamma \vdash \forall x.B} \forall\mathcal{R}, h \notin \Sigma, \text{supp}(B) = \{\vec{c}\} \\
\\
\frac{\Sigma; \Gamma, B[a/x] \vdash C}{\Sigma; \Gamma, \nabla x.B \vdash C} \nabla\mathcal{L}, a \notin \text{supp}(B) \quad \frac{\Sigma; \Gamma \vdash B[a/x]}{\Sigma; \Gamma \vdash \nabla x.B} \nabla\mathcal{R}, a \notin \text{supp}(B) \\
\\
\frac{\Sigma, h; \Gamma, B[h\vec{c}/x] \vdash C}{\Sigma; \Gamma, \exists x.B \vdash C} \exists\mathcal{L}, h \notin \Sigma, \text{supp}(B) = \{\vec{c}\} \quad \frac{\Sigma, \mathcal{K}, \mathcal{C}_{\mathcal{N}} \vdash t : \tau \quad \Sigma; \Gamma \vdash B[t/x]}{\Sigma; \Gamma \vdash \exists x.B} \exists\mathcal{R}
\end{array}$$

Fig. 1. The inference rules of LG

We now extend the logic LG with a proof theoretic notion of equality and fixed points, following on works by Hallnas and Schroeder-Heister [2,12], Girard [1] and McDowell and Miller [3]. The equality rules are as follows:

$$\frac{\{\Sigma\theta; \Gamma\theta \vdash C\theta \mid (\lambda\vec{c}.t)\theta =_{\beta\eta} (\lambda\vec{c}.s)\theta\}}{\Sigma; \Gamma, s = t \vdash C} eq\mathcal{L} \quad \frac{}{\Sigma; \Gamma \vdash t = t} eq\mathcal{R}$$

where $\text{supp}(s = t) = \{\vec{c}\}$ in the $eq\mathcal{L}$ rule. In the $eq\mathcal{L}$ rule, the substitution θ is a *unifier* of $\lambda\vec{c}.s$ and $\lambda\vec{c}.t$. We specify the premise of the rule as a set to mean that every element of the set is a premise. Since the terms s and t can be arbitrary higher-order terms, in general the set of their unifiers can be infinite. However, in some restricted cases, e.g., when $\lambda\vec{c}.s$ and $\lambda\vec{c}.t$ are *higher-order pattern* terms [5,9], if both terms are unifiable, then there exists a most general unifier. The applications we are considering are those which satisfy the higher-order pattern restrictions.

Definition 2. To each atomic formula, we associate a fixed point equation, or a definition clause, following the terminology of $FO\lambda^{\Delta\nabla}$. A definition clause is written $\forall\vec{x}.p\vec{x} \triangleq B$ where the free variables of B are among \vec{x} . The predicate $p\vec{x}$ is called the head of the definition clause, and B is called the body. A definition is a set of definition clauses. We often omit the outer quantifiers when referring to a definition clause.

The introduction rules for defined atoms are as follows:

$$\frac{\Sigma; \Gamma, B[\vec{t}/\vec{x}] \vdash C}{\Sigma; \Gamma, p\vec{t} \vdash C} def\mathcal{L}, p\vec{x} \triangleq B \quad \frac{\Sigma; \Gamma \vdash B[\vec{t}/\vec{x}]}{\Sigma; \Gamma \vdash p\vec{t}} def\mathcal{R}, p\vec{x} \triangleq B$$

In order to prove the cut-elimination theorem and the consistency of LG^{ω} , we allow only definition clauses which satisfy an *equivariance preserving* condition and a certain positivity condition, so as to guarantee the existence of fixed points.

Definition 3. We associate with each predicate symbol p a natural number, the level of p . Given a formula B , its level $lvl(B)$ is defined as follows:

1. $lvl(p\bar{t}) = lvl(p)$
2. $lvl(\perp) = lvl(\top) = 0$
3. $lvl(B \wedge C) = lvl(B \vee C) = \max(lvl(B), lvl(C))$
4. $lvl(B \supset C) = \max(lvl(B) + 1, lvl(C))$
5. $lvl(\forall x.B) = lvl(\nabla x.B) = lvl(\exists x.B) = lvl(B)$.

A definition clause $p\bar{x} \triangleq B$ is stratified if $lvl(B) \leq lvl(p)$ and B has no free occurrences of nominal constants. We consider only definition clauses which are stratified.

An example that violates the first restriction in Definition 3 is the definition $p \triangleq p \supset \perp$. In [12], Schroeder-Heister shows that admitting this definition in a logic with contraction leads to inconsistency. To see why we need the second restriction on name constants, consider the definition $qx \triangleq (x = a)$, where a is a nominal constant. Let b be a nominal constant different from a . Using this definition, we would be able to derive \perp :

$$\frac{\frac{\frac{}{\vdash a = a} \text{eq}\mathcal{R}}{\vdash qa} \text{def}\mathcal{R} \quad \frac{\frac{\frac{}{qa \vdash qb} \text{id}_\pi \quad \frac{\frac{}{b = a \vdash \perp} \text{eq}\mathcal{L}}{qb \vdash \perp} \text{def}\mathcal{L}}{qa \vdash \perp} \text{cut}}{\vdash \perp} \text{cut}}{\vdash \perp} \text{cut}$$

In examples and applications, we often express definition clauses with patterns in the heads. Let us consider, for example, a definition clause for lists. We first introduce a type lst to denote lists of elements of type α , and the constants

$$nil : lst \quad :: \quad : \alpha \rightarrow lst \rightarrow lst$$

which denote the empty list and a constructor to build a list from an element of type α and another list. The latter will be written in the infix notation. The definition clause for *lists* is as follows.

$$list\ L \triangleq L = nil \vee \exists_\alpha A \exists_{lst} L'. L = (A :: L') \wedge list\ L'.$$

Using patterns, the above definition of lists can be rewritten as

$$list\ nil \triangleq \top. \quad list\ (A :: L) \triangleq list\ L.$$

We shall often work directly with this patterned notation for definition clauses. For this purpose, we introduce the notion of *patterned definitions*. A *patterned definition clause* is written $\forall \vec{x}. H \triangleq B$ where the free variables of H and B are among \vec{x} . The stratification of definitions in Definition 3 applies to patterned definitions as well. Since the patterned definition clauses are not allowed to have free occurrences of nominal constants, in matching the heads of the clauses with an atomic formula in a sequent, we need to raise the variables of the clauses to account for nominal constants that are in the support of the introduced formula. Given a patterned definition clause $\forall x_1 \dots \forall x_n. H \triangleq B$ its raised clause with respect to the list of constants $c_1 : \iota_1 \dots c_n : \iota_n$ is

$$\forall h_1 \dots \forall h_n. H[h_1 \ \vec{c}/x_1, \dots, h_n \ \vec{c}/x_n] \triangleq B[h_1 \ \vec{c}/x_1, \dots, h_n \ \vec{c}/x_n].$$

The introduction rules for patterned definitions are

$$\frac{\{\Sigma\theta; B\theta, \Gamma\theta \vdash C\theta\}_\theta}{\Sigma; A, \Gamma \vdash C} \text{def}\mathcal{L} \quad \frac{\Sigma; \Gamma \vdash B\theta}{\Sigma; \Gamma \vdash A} \text{def}\mathcal{R}$$

In the $\text{def}\mathcal{L}$ rule, B is the body of the raised patterned clause $\forall x_1 \dots \forall x_n. H \triangleq B$ and $(\lambda \vec{c}. H)\theta = (\lambda \vec{c}. A)\theta$ where $\{\vec{c}\}$ is the support of A . In the $\text{def}\mathcal{R}$ rule, we match A with the head of the clause, i.e., $\lambda \vec{c}. A = (\lambda \vec{c}. H)\theta$. These patterned rules can be derived using the non-patterned definition rules and the equality rules, as shown in [13],

Natural number induction. We introduce a type nt to denote natural numbers, with the usual constants $z : nt$ (zero) and $s : nt \rightarrow nt$ (the successor function), and a special predicate $nat : nt \rightarrow nt \rightarrow o$. The rules for natural number induction are the same as those in $FO\lambda^{\Delta\mathbb{N}}$ [3], which are the introduction rules for the predicate nat .

$$\frac{\vdash Dz \quad j; Dj \vdash D(sj) \quad \Sigma; \Gamma, DI \vdash C}{\Sigma; \Gamma, nat I \vdash C} nat\mathcal{L}$$

$$\frac{}{\Sigma; \Gamma \vdash nat z} nat\mathcal{R} \quad \frac{\Sigma; \Gamma \vdash nat I}{\Sigma; \Gamma \vdash nat(sI)} nat\mathcal{R}$$

The logic LG extended with the equality, definitions and induction rules is referred to as LG^ω .

3 Properties of derivations

In this section we examine several properties of the ∇ -quantifier and derivations in LG^ω that are useful in the cut elimination proof. These properties concern the transformation of derivations, in particular, they state that provability is preserved under Σ -substitutions, permutations and a restricted form of name substitutions.

We first look at the properties of the ∇ quantifier in relation to other connectives. The proof of the following proposition is straightforward by inspection on the rules of LG .

Proposition 4. *The following formulas are provable in LG :*

1. $\nabla x.(Bx \wedge Cx) \equiv \nabla x.Bx \wedge \nabla x.Cx.$
2. $\nabla x.(Bx \supset Cx) \equiv \nabla x.Bx \supset \nabla x.Cx.$
3. $\nabla x.(Bx \vee Cx) \equiv \nabla x.Bx \vee \nabla x.Cx.$
4. $\nabla x.B \equiv B$, provided that x is not free in B .
5. $\nabla x \nabla y.Bxy \equiv \nabla y \nabla x.Bxy.$
6. $\forall x.Bx \supset \nabla x.Bx.$
7. $\nabla x.Bx \supset \exists x.Bx.$

The formulas (1) – (3) are provable in $FO\lambda^\nabla$. The proposition is true also in nominal logic with ∇ replaced by \mathbb{N} .

Definition 5. *Given a derivation Π with premise derivations $\{\Pi_i\}_{i \in \mathcal{I}}$ where \mathcal{I} is some index set, the measure $ht(\Pi)$, the height of Π , is defined as the least upper bound of $\{ht(\Pi_i) + 1\}_{i \in \mathcal{I}}$.*

We now define some transformations of derivations: weakening of hypotheses, substitutions on derivations, permutations and restricted name substitutions. In the following definitions we omit the signatures in the sequents if it is clear from context which signatures we refer to. We denote with id the identity function on \mathcal{C}_N .

Definition 6. *Weakening of hypotheses. Let Π be a derivation of $\Sigma; \Gamma \vdash C$. Let Δ be a multiset of formulas whose free variables are among Σ . We define the derivation $w(\Delta, \Pi)$ of $\Sigma; \Gamma, \Delta \vdash C$ as follows:*

1. *If Π ends with $eq\mathcal{L}$*

$$\frac{\left\{ \frac{\Pi_\theta}{\Sigma\theta; \Gamma'\theta \vdash C\theta} \right\}_\theta}{\Sigma; s = t, \Gamma' \vdash C} eq\mathcal{L}$$

then $w(\Delta, \Pi)$ is

$$\frac{\left\{ \frac{w(\Delta\theta, \Pi_\theta)}{\Sigma\theta; \Gamma'\theta, \Delta\theta \vdash C\theta} \right\}_\theta}{\Sigma; s = t, \Gamma', \Delta \vdash C} eq\mathcal{L}$$

2. If Π ends with $\text{nat}\mathcal{L}$

$$\frac{\frac{\Pi_1}{\vdash D z} \quad \frac{\Pi_2}{D i \vdash D (s i)} \quad \frac{\Pi_3}{D I, \Gamma' \vdash C}}{\text{nat } I, \Gamma' \vdash C} \text{ nat}\mathcal{L}$$

then $w(\Delta, \Pi)$ is

$$\frac{\frac{\Pi_1}{\vdash D z} \quad \frac{\Pi_2}{D i \vdash D (s i)} \quad \frac{w(\Delta, \Pi_3)}{D I, \Gamma', \Delta \vdash C}}{\text{nat } I, \Gamma', \Delta \vdash C} \text{ nat}\mathcal{L}$$

3. If Π ends with the mc rule

$$\frac{\frac{\Pi_1}{\Delta_1 \vdash B_1} \quad \dots \quad \frac{\Pi_n}{\Delta_n \vdash B_n} \quad B_1, \dots, B_n, \Gamma' \vdash C}{\Delta_1, \dots, \Delta_n, \Gamma' \vdash C} \text{ mc}$$

then $w(\Delta, \Pi)$ is

$$\frac{\frac{\Pi_1}{\Delta_1 \vdash B_1} \quad \dots \quad \frac{\Pi_n}{\Delta_n \vdash B_n} \quad B_1, \dots, B_n, \Gamma', \Delta \vdash C}{\Delta_1, \dots, \Delta_n, \Gamma', \Delta \vdash C} \text{ mc}$$

4. If Π ends with any other rule and has premise derivations Π_1, \dots, Π_n then $w(\Delta, \Pi)$ ends with the same rule with premise derivations $w(\Delta, \Pi_1), \dots, w(\Delta, \Pi_n)$.

Definition 7. Substitutions on derivations. If Π is a derivation of $\Sigma; \Gamma \vdash C$ and θ is a Σ -substitution, then we define the derivation $\Pi\theta$ of $\Sigma\theta; \Gamma\theta \vdash C\theta$ as follows:

1. Suppose Π ends with $\text{eq}\mathcal{L}$:

$$\frac{\left\{ \frac{\Pi_\rho}{\Sigma\rho; \Gamma'\rho \vdash C\rho} \right\}_\rho}{\Sigma; s = t, \Gamma' \vdash C} \text{ eq}\mathcal{L}$$

where each ρ is a unifier of $\lambda\vec{c}.s$ and $\lambda\vec{c}.t$. Observe that if ρ' is a unifier of $(\lambda\vec{c}.s)\theta$ and $(\lambda\vec{c}.t)\theta$, then $\theta \circ \rho'$ is a unifier of $\lambda\vec{c}.s$ and $\lambda\vec{c}.t$. Thus $\Pi\theta$ is the derivation:

$$\frac{\left\{ \frac{\Pi_{\theta \circ \rho'}}{\Sigma\theta\rho'; \Delta\theta\rho \vdash C\theta\rho} \right\}_{\rho'}}{\Sigma; s\theta = t\theta, \Delta\theta \vdash C\theta} \text{ eq}\mathcal{L}$$

2. Suppose Π ends with $\forall\mathcal{R}$:

$$\frac{\frac{\Pi_1}{\Sigma; \Gamma \vdash B[h\vec{c}/x]}}{\Sigma; \Gamma \vdash \forall x.B} \forall\mathcal{R},$$

where $\{\vec{c}\} = \text{supp}(\forall x.B)$. Let $\{\vec{d}\}$ be the support of $(\forall x.B)\theta$, which might be smaller than $\{\vec{c}\}$. Let ρ be the substitution $[\lambda\vec{c}.h'\vec{d}/h]$ where h' is a new variable not already in Σ and not among the free variables in θ . We can assume without loss of generality that x is not free in θ , hence $((B[h\vec{c}/x])\rho)\theta = (B[h'\vec{d}/x])\theta = (B\theta)[h'\vec{d}/x]$. Then $\Pi\theta$ is

$$\frac{\frac{\Pi_1(\rho \circ \theta)}{\Sigma\theta, h'; \Gamma\theta \vdash (B\theta)[h'\vec{d}/x]}}{\Sigma\theta; \Gamma\theta \vdash (\forall x.B)\theta} \forall\mathcal{R}$$

3. Suppose Π ends with $\exists\mathcal{L}$: this case is dual to the previous one.

4. If Π ends with any other rule and has premise derivations Π_1, \dots, Π_n , then $\Pi\theta$ ends with the same rule and has premise derivations $\Pi_1\theta, \dots, \Pi_n\theta$.

Definition 8. Let Π be a proof of $\Sigma; B_1, \dots, B_n \vdash B_0$ and let $\vec{\pi} = \pi_0, \dots, \pi_n$ be a list of permutations. We define a derivation $\langle \vec{\pi} \rangle. \Pi$ of $\Sigma; \pi_1.B_1, \dots, \pi_n.B_n \vdash \pi_0.B_0$ as follows:

1. Suppose that Π ends with id_π

$$\frac{\pi.B_j = \pi'.B_0}{\Sigma; B_1, \dots, B_n \vdash B_0} \text{id}_\pi.$$

Obverse that $\pi.\pi_j^{-1}.\pi_j.B = \pi'.\pi_0^{-1}.\pi_0.B'$. Hence $\langle \vec{\pi} \rangle. \Pi$ ends with the same rule.

2. Suppose Π ends with mc :

$$\frac{\frac{\Pi_1}{\Delta_1 \vdash D_1} \dots \frac{\Pi_m}{\Delta_m \vdash D_m} \quad D_1, \dots, D_m, \Delta_{m+1} \vdash B_0}{B_1, \dots, B_n \vdash B_0} \text{mc}$$

where $\Delta_1, \dots, \Delta_{m+1}$ are partitions of B_1, \dots, B_n . Suppose that for each $i \in \{1, \dots, m+1\}$, $\Delta_i = B_{i1}, \dots, B_{ik_i}$ for some index k_i . Let $\vec{\pi}(i)$, for $i \in \{1, \dots, m\}$, be the permutations $\text{id}, \pi_{i1}, \dots, \pi_{ik_i}$. Let $\vec{\pi}(m+1)$ be the permutations

$$\pi_0, \underbrace{\text{id}, \dots, \text{id}}_m, \pi_{(m+1)1}, \dots, \pi_{(m+1)k_{m+1}}$$

We denote with Δ'_i the list

$$\pi_{i1}.B_{i1}, \dots, \pi_{ik_i}.B_{ik_i}.$$

Then $\langle \vec{\pi} \rangle. \Pi$ is the derivation

$$\frac{\frac{\langle \vec{\pi}(1) \rangle. \Pi_1}{\Delta'_1 \vdash D_1} \dots \frac{\langle \vec{\pi}(m) \rangle. \Pi_m}{\Delta'_m \vdash D_m} \quad D_1, \dots, D_m, \Delta'_{m+1} \vdash \pi_0.B_0}{\pi_1.B_1, \dots, \pi_n.B_n \vdash \pi_0.B_0} \text{mc}$$

3. Suppose Π ends with $\nabla \mathcal{R}$:

$$\frac{\Pi_1}{\Sigma; B_1, \dots, B_n \vdash B[a/x]} \nabla \mathcal{R}$$

where $a : \iota \notin \text{supp}(B)$. Let $d : \iota$ be a nominal constant such that $d \notin \text{supp}(B)$ and $\pi_0(d) = d$. Such a constant exists since $\text{supp}(B)$ is finite and π_0 is a finite permutation. Thus $\pi_0.(a \ d).B_0[a/x] = \pi_0.B_0[d/x]$. Then $\langle \vec{\pi} \rangle. \Pi$ is the derivation:

$$\frac{\langle \pi_0.(a \ d), \dots, \pi_n \rangle. \Pi_1}{\Sigma; \pi_1.B_1, \dots, \pi_n.B_n \vdash \pi_0.B[d/x]} \nabla \mathcal{R}$$

4. Suppose Π ends with $\nabla \mathcal{L}$: this case is analogous to previous one.
5. Suppose Π ends with $c\mathcal{L}$:

$$\frac{\Pi'}{B_1, \dots, B_j, B_j, \dots, B_n \vdash B_0} c\mathcal{L}$$

then $\langle \vec{\pi} \rangle. \Pi$ is

$$\frac{\langle \pi_1, \dots, \pi_j, \pi_j, \dots, \pi_n \rangle. \Pi'}{\pi_1.B_1, \dots, \pi_j.B_j, \pi_j.B_j, \dots, \pi_n.B_n \vdash \pi_0.B_0} c\mathcal{L}$$

6. If Π ends with any other rule and has premise derivations Π_1, \dots, Π_m , then $\langle \vec{\pi} \rangle. \Pi$ ends with the same rule and has premise derivations $\langle \vec{\pi} \rangle. \Pi_1, \dots, \langle \vec{\pi} \rangle. \Pi_m$.

Definition 9. Let Π be a proof of $\Sigma, x : \iota; B_1, \dots, B_n \vdash B_0$ and let $\vec{a} = a_0, \dots, a_n$ be a list of nominal constants such that $a_i \notin \text{supp}(B_i)$. We define a derivation $r(x, \langle \vec{a} \rangle, \Pi)$ of $\Sigma; B_1[a_1/x], \dots, B_n[a_n/x] \vdash B_0[a_0/x]$, as follows:

1. Suppose Π is

$$\frac{\pi.B_j = \pi'.B_0}{\Sigma, x; B_1, \dots, B_n \vdash B_0} id_\pi.$$

Let $d : \iota$ be a nominal constant which is not in the support of B_j and B_0 , and $\pi(d) = d$ and $\pi'(d) = d$. Then $r(x, \vec{a}, \Pi)$ is

$$\frac{\pi.(a_j \ d).B_1[a_1/x] = \pi'.(a_0 \ d).B_0[a_0/x]}{\Sigma; B_1[a_1/x], \dots, B_n[a_n/x] \vdash B_0[a_0/x]} id_\pi$$

2. Suppose Π ends with mc :

$$\frac{\begin{array}{c} \Pi_1 \\ \Sigma, x; \Delta_1 \vdash D_1 \end{array} \quad \dots \quad \begin{array}{c} \Pi_m \\ \Sigma, x; \Delta_m \vdash D_m \end{array} \quad \begin{array}{c} \Pi' \\ \Sigma, x; D_1, \dots, D_m, \Delta_{m+1} \vdash B_0 \end{array}}{\Sigma, x; B_1, \dots, B_n \vdash B_0} mc$$

where $\Delta_1, \dots, \Delta_{m+1}$ is a partition of B_1, \dots, B_n . Suppose that for each $i \in \{1, \dots, m+1\}$, $\Delta_i = B_{i1}, \dots, B_{ik_i}$ for some index k_i . Let $\vec{d} = d_1, \dots, d_m$ be a list of nominal constants such that $d_i \notin \text{supp}(D_i)$. Let $f(i)$, for $i \in \{1, \dots, m\}$ be the list $d_i, a_{i1}, \dots, a_{ik_i}$ and let $f(m+1)$ be the list

$$a_0, \vec{d}, a_{(m+1)1}, \dots, a_{(m+1)k_{(m+1)}}.$$

Let Δ'_i be the list

$$B_{i1}[a_{i1}/x], \dots, B_{ik_i}[a_{ik_i}/x]$$

and let Γ be the list

$$D_1[d_1/x], \dots, D_m[d_m/x], \Delta'_{m+1}.$$

Then $r(x, \vec{a}, \Pi)$ is the derivation

$$\frac{\begin{array}{c} r(x, f(1), \Pi_1) \\ \Sigma; \Delta'_1 \vdash D_1[d_1/x] \end{array} \quad \dots \quad \begin{array}{c} r(x, f(m), \Pi_m) \\ \Sigma; \Delta'_m \vdash D_m[d_m/x] \end{array} \quad \begin{array}{c} r(x, f(m+1), \Pi') \\ \Sigma; \Gamma \vdash B_0[a_0/x] \end{array}}{\Sigma; B_1[a_1/x], \dots, B_n[a_n/x] \vdash B_0[a_0/x]} mc$$

3. Suppose Π is

$$\frac{\Pi_1}{\Sigma, x; B_1, \dots, B_n \vdash B[c/y]} \nabla \mathcal{R}.$$

If $a_0 \neq c$ then $r(x, \vec{a}, \Pi)$ is

$$\frac{\begin{array}{c} r(x, \vec{a}, \Pi_1) \\ \Sigma, x; B_1, \dots, B_n \vdash B[c/y] \end{array}}{\Sigma, x; B_1, \dots, B_n \vdash \nabla y.B} \nabla \mathcal{R}.$$

If $a_0 = c$, then we swap c with a fresh constant. Let $d : \iota$ be a nominal constant not in the support of $B[c/y]$. We apply the swapping ($c \ d$) to the conclusion of the end sequent of Π_1 according to the construction in Definition 8 to get a proof Π_2 of $\Sigma, x; B_1, \dots, B_n \vdash B_0[d/y]$. The derivation $r(x, \vec{a}, \Pi)$ is constructed as follows:

$$\frac{\begin{array}{c} r(x, \vec{a}, \Pi_2) \\ \Sigma; B_1[a_1/x], \dots, B_n[a_n/x] \vdash B[a_0/x, d/y] \end{array}}{\Sigma; B_1[a_1/x], \dots, B_n[a_n/x] \vdash \nabla y.B[a_0/x]} \nabla \mathcal{R}$$

4. If Π ends with $\nabla \mathcal{L}$ apply the same construction as in the previous case.

5. Suppose Π ends with $\forall\mathcal{R}$

$$\frac{\Pi_1}{\frac{\Sigma, x, h; B_1, \dots, B_n \vdash B[h\vec{c}/y]}{\Sigma, x; B_1, \dots, B_n \vdash \forall y.B} \forall\mathcal{R}} .$$

Let $\theta = [\lambda\vec{c}.h'\vec{c}x/h]$ where h' is a variable not in Σ . Apply the construction in Definition 7 to get the proof $\Pi\theta$ of

$$\Sigma, x, h'; B_1, \dots, B_n \vdash B[h'\vec{c}x/y]$$

Then $r(x, \vec{a}, \Pi)$ is

$$\frac{r(x, \vec{a}, \Pi\theta)}{\frac{\Sigma, h'; B_1[a_1/x], \dots, B_n[a_n/x] \vdash B[a_0/x, (h'\vec{c}a_0)/y]}{\Sigma; B_1[a_1/x], \dots, B_n[a_n/x] \vdash \forall y.B[a_0/x]} \forall\mathcal{R}} .$$

6. If Π ends with $\exists\mathcal{L}$, apply the same construction as in the previous case.

7. Suppose Π ends with $\exists\mathcal{R}$:

$$\frac{\Pi_1}{\frac{\Sigma, x; B_1, \dots, B_n \vdash B[t/y]}{\Sigma, x; B_1, \dots, B_n \vdash \exists y.B} \exists\mathcal{R}} .$$

If $a_0 \notin \text{supp}(B[t/y])$ then $r(x, \vec{a}, \Pi)$ is

$$\frac{r(x, \vec{a}, \Pi_1)}{\frac{\Sigma; B_1[a_1/x], \dots, B_n[a_n/x] \vdash B[a_0/x, t/y]}{\Sigma; B_1[a_1/x], \dots, B_n[a_n/x] \vdash \exists y.B[a_0/x]} \exists\mathcal{R}} .$$

If $a_0 \in \text{supp}(B[t/y])$, we exchange it with a fresh constant. Let d be a nominal constant distinct from a_0 and not in the support of $B[t/y]$. Then $((a_0 d).B[t/y])[a_0/x] = B[(a_0 d).t/y, a_0/x]$. We first apply the construction in Definition 8 to Π_1 to get a derivation Π_2 of $\Sigma, x; B_1, \dots, B_n \vdash B[(a_0 d).t/y, a_0/x]$. The derivation $r(x, \vec{a}, \Pi)$ is thus

$$\frac{r(x, \vec{a}, \Pi_2)}{\frac{\Sigma; B_1[a_1/x], \dots, B_n[a_n/x] \vdash B[(a_0 d).t/y, a_0/x]}{\Sigma; B_1[a_1/x], \dots, B_n[a_n/x] \vdash \exists y.B[a_0/x]} \exists\mathcal{R}} .$$

8. Suppose Π ends with $\text{eq}\mathcal{L}$:

$$\frac{\left\{ \frac{\Pi_\theta}{(\Sigma, x)\theta; B_2\theta, \dots, B_n\theta \vdash B_0\theta} \right\}_\theta}{\Sigma, x; s = t, B_2, \dots, B_n \vdash B_0} \text{eq}\mathcal{L}$$

where each θ is a unifier of $(\lambda\vec{c}.s, \lambda\vec{c}.t)$ and $\{\vec{c}\} = \text{supp}(s = t)$. We need to show that for each unifier of $(\lambda a_1 \lambda\vec{c}.s[a_1/x], \lambda a_1 \lambda\vec{c}.t[a_1/x])$ there is a corresponding unifier for $\lambda\vec{c}.s$ and $\lambda\vec{c}.t$. We can assume without loss of generality that x is not in the domain of ρ .

We first show the case where x is not free in ρ . It is clear that in this case ρ is a unifier of $\lambda\vec{c}.s$ and $\lambda\vec{c}.t$. Therefore we apply the procedure recursively to the premise derivation Π_ρ , to get the derivation $r(x, \vec{a}, \Pi_\rho)$ of

$$\Sigma\rho; (B_2[a_2/x])\rho, \dots, (B_n[a_n/x])\rho \vdash (B_0[a_0/x])\rho.$$

In the other case, where x is free in the range of ρ , we show that it can be reduced to the previous case. First we define a substitution ρ' to be the substitution ρ where x is replaced by a new variable u which is not free in ρ . Clearly ρ' is also a unifier of $\lambda a_1 \lambda\vec{c}.s[a_1/x]$ and $\lambda a_1 \lambda\vec{c}.t[a_1/x]$. Moreover, it is more general than ρ , since $\rho = [x/u] \circ \rho'$. Therefore we can apply the construction in the previous case to get a derivation $r(x, \vec{a}, \Pi_{\rho'})$ and apply the substitution $[x/u]$ to this derivation, using the procedure in Definition 7, to get a derivation of

$$\Sigma\rho; (B_2[a_2/x])\rho, \dots, (B_n[a_n/x])\rho \vdash (B_0[a_0/x])\rho.$$

The derivation $r(x, \vec{a}, \Pi)$ is then constructed as follows

$$\frac{\left\{ \frac{\Pi'_\rho}{\Sigma\rho; (B_2[a_2/x])\rho, \dots, (B_n[a_n/x])\rho \vdash (B_0[a_0/x])\rho} \right\}_\rho}{\Sigma; s[a_1/x] = t[a_1/x], \dots, B_n[a_n/x] \vdash B_0[a_0/x]} \text{eq}\mathcal{L}$$

where each Π'_ρ is constructed as explained above.

9. If Π ends with $c\mathcal{L}$:

$$\frac{B_1, \dots, B_j, B_j, \dots, B_n \vdash B_0}{B_1, \dots, B_j, \dots, B_n \vdash B_0} \frac{\Pi'}{c\mathcal{L}}$$

then $r(x, \vec{a}, \Pi)$ is

$$\frac{r(x, (a_0, \dots, a_j, a_j, \dots, a_n), \Pi')}{B_1[a_1/x], \dots, B_j[a_j/x], B_j[a_j/x], \dots, B_n[a_n/x] \vdash B_0[a_0/x]} \frac{B_1[a_1/x], \dots, B_j[a_j/x], B_j[a_j/x], \dots, B_n[a_n/x] \vdash B_0[a_0/x]}{B_1[a_1/x], \dots, B_j[a_j/x], \dots, B_n[a_n/x] \vdash B_0[a_0/x]} c\mathcal{L}$$

10. If Π ends with any other rule and has premise derivations Π_1, \dots, Π_n , then $r(x, \vec{a}, \Pi)$ ends with the same rule and has premise derivations $r(x, \vec{a}, \Pi_1), \dots, r(x, \vec{a}, \Pi_n)$.

Lemma 10. For any derivation Π of $\Sigma; \Gamma \vdash C$ and any multiset of Σ -formulas Δ , $w(\Delta, \Pi)$ is a derivation of $\Sigma; \Gamma, \Delta \vdash C$ and $ht(w(\Delta, \Pi)) \leq ht(\Pi)$.

Lemma 11. For any derivation Π of $\Sigma; \Gamma \vdash C$ and any Σ -substitution θ , $\Pi\theta$ is a derivation of $\Sigma\theta; \Gamma\theta \vdash C\theta$ and $ht(\Pi\theta) \leq ht(\Pi)$.

Lemma 12. For any derivation Π of $B_1, \dots, B_n \vdash B_0$ and permutations $\vec{\pi} = \pi_0, \dots, \pi_n$, $\langle \vec{\pi} \rangle. \Pi$ is a derivation of $\pi_1.B_1, \dots, \pi_n.B_n \vdash \pi_0.B_0$ and $ht(\langle \vec{\pi} \rangle. \Pi) \leq ht(\Pi)$.

Lemma 13. For any derivation Π of $\Sigma, x; B_1, \dots, B_n \vdash B_0$ and any list of nominal constants $\vec{a} = a_0, \dots, a_n$ such that $a_i \notin \text{supp}(B_i)$, $r(x, \vec{a}, \Pi)$ is a derivation of $\Sigma; B_1[a_1/x], \dots, B_n[a_n/x] \vdash B_0[a_0/x]$ and $ht(r(x, \vec{a}, \Pi)) \leq ht(\Pi)$.

Lemma 14. Substitutions. Let Π be a proof of $\Sigma; \Gamma \vdash C$ and let θ be a Σ -substitution. Then there exists a proof Π' of $\Sigma\theta; \Gamma\theta \vdash C\theta$ such that $ht(\Pi') \leq ht(\Pi)$.

Proof. Follows immediately from Lemma 11. □

Lemma 15. Permutations. Let Π be a proof of $\Sigma; B_1, \dots, B_n \vdash B_0$. Then there exists a proof Π' of $\Sigma; \pi_1.B_1, \dots, \pi_n.B_n \vdash \pi_0.B_0$ such that $ht(\Pi') \leq ht(\Pi)$.

Proof. Follows immediately from Lemma 12. □

Lemma 16. Restricted name substitutions. Let Π be a proof of

$$\Sigma, x : \iota; B_1, \dots, B_n \vdash B_0.$$

Then there exists a proof of Π' of $\Sigma; B_1[a_1/x], \dots, B_n[a_n/x] \vdash B_0[a_0/x]$, where $a_i \notin \text{supp}(B_i)$ for each $i \in \{0, \dots, n\}$, such that $ht(\Pi') \leq ht(\Pi)$.

Proof. Follows immediately from Lemma 13. □

The next two lemmas are crucial to the cut-elimination proof: they allow one to reintroduce the symmetry between $\forall\mathcal{L}$ and $\forall\mathcal{R}$, and dually, between $\exists\mathcal{L}$ and $\exists\mathcal{R}$ rules.

Lemma 17. Support extension. Let Π be a proof of $\Sigma, h; \Gamma \vdash B[h \vec{a}/x]$ where $\{\vec{a}\} = \text{supp}(B)$, $h \notin \Sigma$ and h is not free in Γ and B . Let \vec{c} be a list of nominal constants not in the support of B . Then there exists a proof Π' of $\Sigma, h'; \Gamma \vdash B[h' \vec{a}\vec{c}/x]$ where $h' \notin \Sigma$.

Proof. Suppose \vec{c} is the list of constants $c_1 : \iota_1, \dots, c_n : \iota_n$. Let $\vec{y} = y_1 : \iota_1, \dots, y_n : \iota_n$ be a list of distinct variables not appearing in $\Sigma \cup \{h, h'\}$. We first apply the substitution $[\lambda \vec{a}. h' \vec{c}\vec{x}/h]$ to the sequent $\Sigma, h; \Gamma \vdash B[h\vec{a}/x]$. By Lemma 14, there is a proof Π_1 of

$$\Sigma, h', \vec{y}; \Gamma \vdash B[h' \vec{a}\vec{y}/x]$$

The derivation Π' is then obtained by repeatedly applying Lemma 16 to Π_1 to change \vec{y} into \vec{c} . \square

Lemma 18. Support extension. Let Π be a proof of $\Sigma, h; B[h \vec{a}/x], \Gamma \vdash C$ where $\{\vec{a}\} = \text{supp}(B)$, $h \notin \Sigma$ and h is not free in Γ , B and C . Let \vec{c} be a list of nominal constants not in the support of B . Then there exists a proof Π' of $\Sigma, h'; B[h' \vec{a}\vec{c}/x], \Gamma \vdash C$ where $h' \notin \Sigma$.

Proof. Use the same construction as in the proof of Lemma 17. \square

4 Cut reduction

We define a *reduction* relation between derivations, following closely the reduction relation in [3]. For simplicity of presentation, we shall omit the signatures in the sequents in the following reduction of cuts when the signatures are not changed by the reduction or when it is clear from context which signatures should be assigned to the sequents. The redex is always a derivation Ξ ending with the multicut rule

$$\frac{\Sigma; \Delta_1 \vdash B_1 \quad \dots \quad \Sigma; \Delta_n \vdash B_n \quad \Sigma; B_1, \dots, B_n, \Gamma \vdash C}{\Sigma; \Delta_1, \dots, \Delta_n, \Gamma \vdash C} mc.$$

We refer to the formulas B_1, \dots, B_n produced by the mc as *cut formulas*.

If $n = 0$, Ξ reduces to the premise derivation Π .

For $n > 0$ we specify the reduction relation based on the last rule of the premise derivations. If the rightmost premise derivation Π ends with a left rule acting on a cut formula B_i , then the last rule of Π_i and the last rule of Π together determine the reduction rules that apply. We classify these rules according to the following criteria: we call the rule an *essential* case when Π_i ends with a right rule; if it ends with a left rule, it is a *left-commutative* case; if Π_i ends with the *id* rule, then we have an *axiom* case; a *multicut* case arises when it ends with the mc rule. When Π does not end with a left rule acting on a cut formula, then its last rule is alone sufficient to determine the reduction rules that apply. If Π ends in a rule acting on a formula other than a cut formula, then we call this a *right-commutative* case. A *structural* case results when Π ends with a contraction or weakening on a cut formula. If Π ends with the *id* rule, this is also an axiom case; similarly a multicut case arises if Π ends in the mc rule.

For simplicity of presentation, we always show $i = 1$.

Essential cases:

$\wedge\mathcal{R}/\wedge\mathcal{L}$: If Π_1 and Π are

$$\frac{\frac{\Pi_1' \quad \Pi_1''}{\Delta_1 \vdash B_1' \quad \Delta_1 \vdash B_1''} \wedge\mathcal{R}}{\Delta_1 \vdash B_1' \wedge B_1''} \wedge\mathcal{R} \quad \frac{\frac{\Pi'}{B_1', B_2, \dots, B_n, \Gamma \vdash C}}{B_1' \wedge B_1'', B_2, \dots, B_n, \Gamma \vdash C} \wedge\mathcal{L},$$

then Ξ reduces to

$$\frac{\frac{\Pi_1' \quad \Pi_2 \quad \dots \quad \Pi_n}{\Delta_1 \vdash B_1' \quad \Delta_2 \vdash B_2 \quad \dots \quad \Delta_n \vdash B_n} \quad \frac{\Pi'}{B_1', B_2, \dots, B_n, \Gamma \vdash C}}{\Delta_1, \dots, \Delta_n, \Gamma \vdash C} mc.$$

The case for the other $\wedge\mathcal{L}$ rule is symmetric.

$\vee\mathcal{R}/\vee\mathcal{L}$: If Π_1 and Π are

$$\frac{\frac{\Pi'_1}{\Delta_1 \vdash B'_1} \vee\mathcal{R} \quad \frac{\frac{\Pi'}{B'_1, B_2, \dots, B_n, \Gamma \vdash C} \quad \frac{\Pi''}{B'_1, B_2, \dots, B_n, \Gamma \vdash C}}{B'_1 \vee B'_1, B_2, \dots, B_n, \Gamma \vdash C} \vee\mathcal{L} ,$$

then Ξ reduces to

$$\frac{\frac{\Pi'_1}{\Delta_1 \vdash B'_1} \quad \frac{\Pi_2}{\Delta_2 \vdash B_2} \quad \dots \quad \frac{\Pi_n}{\Delta_n \vdash B_n} \quad \frac{\Pi'}{B'_1, B_2, \dots, B_n, \Gamma \vdash C}}{\Delta_1, \dots, \Delta_n, \Gamma \vdash C} mc .$$

The case for the other $\vee\mathcal{R}$ rule is symmetric.

$\supset\mathcal{R}/\supset\mathcal{L}$: Suppose Π_1 and Π are

$$\frac{\frac{\Pi'_1}{B'_1, \Delta_1 \vdash B''_1} \supset\mathcal{R} \quad \frac{\frac{\Pi'}{B_2, \dots, B_n, \Gamma \vdash B'_1} \quad \frac{\Pi''}{B'_1, B_2, \dots, B_n, \Gamma \vdash C}}{B'_1 \supset B'_1, B_2, \dots, B_n, \Gamma \vdash C} \supset\mathcal{L} .$$

Let Ξ_1 be

$$\frac{\left\{ \frac{\Pi_i}{\Delta_i \vdash B_i} \right\}_{i \in \{2..n\}} \quad \frac{\frac{\Pi'}{B_2, \dots, B_n, \Gamma \vdash B'_1} mc}{\Delta_2, \dots, \Delta_n, \Gamma \vdash B'_1} mc \quad \frac{\Pi'_1}{B'_1, \Delta_1 \vdash B''_1} mc}{\Delta_1, \dots, \Delta_n, \Gamma \vdash B''_1} mc .$$

Then Ξ reduces to

$$\frac{\frac{\Xi_1}{\dots \vdash B''_1} \quad \frac{\left\{ \frac{\Pi_i}{\Delta_i \vdash B_i} \right\}_{i \in \{2..n\}} \quad \frac{\Pi''}{B'_1, \{B_i\}_{i \in \{2..n\}}, \Gamma \vdash C} mc}{\frac{\Delta_1, \dots, \Delta_n, \Gamma, \Delta_2, \dots, \Delta_n, \Gamma \vdash C}{\Delta_1, \dots, \Delta_n, \Gamma \vdash C} c\mathcal{L}} mc .$$

We use the double horizontal lines to indicate that the relevant inference rule (in this case, $c\mathcal{L}$) may need to be applied zero or more times.

$\forall\mathcal{R}/\forall\mathcal{L}$: Suppose Π_1 and Π are

$$\frac{\frac{\Pi'_1}{\Sigma, h; \Delta_1 \vdash B'_1[(h \vec{c})/x]} \forall\mathcal{R} \quad \frac{\frac{\Pi'}{\Sigma; B'_1[t/x], B_2, \dots, B_n, \Gamma \vdash C}}{\Sigma; \forall x. B'_1, B_2, \dots, B_n, \Gamma \vdash C} \forall\mathcal{L} ,$$

where $\{\vec{c}\} = \text{supp}(B'_1)$. Let $\{\vec{d}\} = \text{supp}(B'_1[t/x]) \setminus \text{supp}(B'_1)$. Apply Lemma 17 to get a derivation Π''_1 of $\Sigma, h'; \Delta_1 \vdash B'_1[(h \vec{c} \vec{d})/x]$. The derivation Ξ reduces to

$$\frac{\frac{\Pi''_1[\lambda \vec{c} \vec{d}. t/h']}{\Sigma; \Delta_1 \vdash B'_1[t/x]} \quad \left\{ \frac{\Pi_i}{\Sigma; \Delta_i \vdash B_i} \right\}_{i \in \{2..n\}} \quad \dots \vdash C}{\Sigma; \Delta_1, \dots, \Delta_n, \Gamma \vdash C} mc .$$

$\exists\mathcal{R}/\exists\mathcal{L}$: Suppose Π_1 and Π are

$$\frac{\frac{\Pi'_1}{\Sigma; \Delta_1 \vdash B'_1[t/x]} \exists\mathcal{R}}{\Sigma; \Delta_1 \vdash \exists x.B'_1} \quad \frac{\frac{\Pi'}{\Sigma; h; B'_1[(h\vec{c})/x], B_2, \dots, B_n, \Gamma \vdash C} \exists\mathcal{L}}{\Sigma; \exists x.B'_1, B_2, \dots, B_n, \Gamma \vdash C} \quad ,$$

where $\{\vec{c}\} = \text{supp}(B'_1)$. Let $\{\vec{d}\} = \text{supp}(B'_1[t/x]) \setminus \text{supp}(B'_1)$. Apply Lemma 18 to Π' to get a derivation Π'' of $\Sigma, h'; \Delta_1 \vdash B'_1[(h'\vec{c}\vec{d})/x]$. Then Ξ reduces to

$$\frac{\frac{\Pi'_1}{\Sigma; \Delta_1 \vdash B'_1[t/x]} \quad \dots \quad \frac{\Pi''[\lambda\vec{c}\vec{d}.t/h']}{\Sigma; B'_1[t/x], B_2, \dots, \Gamma \vdash C} \quad mc}{\Sigma; \Delta_1, \dots, \Delta_n, \Gamma \vdash C} .$$

$\nabla\mathcal{R}/\nabla\mathcal{L}$: Suppose Π_1 and Π are

$$\frac{\frac{\Pi'_1}{\Delta_1 \vdash B'_1[a/x]} \nabla\mathcal{R}}{\Delta_1 \vdash \nabla x.B'_1} \quad \frac{\frac{\Pi'}{B'_1[b/x], \dots, B_n, \Gamma \vdash C} \nabla\mathcal{L}}{\nabla x.B'_1, \dots, B_n, \Gamma \vdash C} .$$

Apply the construction in Definition 8 to Π'_1 to swap a with b to get a derivation Π''_1 of $\Delta_1 \vdash B'_1[b/x]$. Ξ reduces to

$$\frac{\frac{\Pi''_1}{\Delta_1 \vdash B'_1[b/x]} \quad \dots \quad \frac{\Pi'}{B'_1[b/x], \dots, B_n, \Gamma \vdash C} \quad mc}{\Delta_1, \dots, \Delta_n, \Gamma \vdash C} .$$

$\text{nat}\mathcal{R}/\text{nat}\mathcal{L}$: Suppose Π_1 is $\overline{\Delta_1 \vdash \text{nat } z}$ $\text{nat}\mathcal{R}$ and Π is

$$\frac{\frac{\Pi'}{\vdash D z} \quad \frac{\Pi''}{D j \vdash D (s j)} \quad \frac{\Pi'''}{D z, B_2, \dots, B_n, \Gamma \vdash C} \quad \text{nat}\mathcal{L}}{\text{nat } z, B_2, \dots, B_n, \Gamma \vdash C} .$$

Then Ξ reduces to

$$\frac{w(\Delta_1, \Pi') \quad \left\{ \frac{\Pi_i}{\Delta_i \vdash B_i} \right\}_{i \in \{2 \dots n\}} \quad \frac{\Pi'''}{D z, B_2, \dots, B_n, \Gamma \vdash C} \quad mc}{\Delta_1, \Delta_2, \dots, \Delta_n, \Gamma \vdash C} .$$

$\text{nat}\mathcal{R}/\text{nat}\mathcal{L}$: Suppose Π_1 is

$$\frac{\frac{\Pi'_1}{\Delta \vdash \text{nat } I}}{\Delta_1 \vdash \text{nat } (s I)} \text{nat}\mathcal{R}$$

and Π is

$$\frac{\frac{\Pi'}{\vdash D z} \quad \frac{\Pi''}{D j \vdash D (s j)} \quad \frac{\Pi'''}{D (s I), B_2, \dots, B_n, \Gamma \vdash C} \quad \text{nat}\mathcal{L}}{\text{nat } (s I), B_2, \dots, B_n, \Gamma \vdash C} .$$

Let Ξ_1 be

$$\frac{\frac{\Pi'_1}{\Delta_1 \vdash \text{nat } I} \quad \frac{\frac{\Pi'}{\vdash D z} \quad \frac{\Pi''}{D j \vdash D (s j)} \quad \overline{D I \vdash D I} \text{ id}_\pi}{\text{nat } I \vdash D I} \quad \text{nat}\mathcal{L}}{\Delta_1 \vdash D I} mc.$$

Suppose $\{\vec{c}\} = \text{supp}(I)$. We apply the procedures in Definition 7 and Definition 9 to Π'' to obtain the derivation Π^\bullet of

$$h; D(h\vec{c}) \vdash D(s(h\vec{c})).$$

Let Ξ_2 be

$$\frac{\Xi_1 \quad \Pi^\bullet[\lambda\vec{c}.I/h]}{\Delta_1 \vdash DI \quad DI \vdash D(sI)} mc.$$

Then Ξ reduces to

$$\frac{\Xi_2 \quad \Pi_2 \quad \dots \quad \Pi_n \quad D(sI), B_2, \dots, B_n, \Gamma \vdash C}{\Delta_1, \dots, \Delta_n, \Gamma \vdash C} mc.$$

$\text{eq}\mathcal{L}/\text{eq}\mathcal{R}$: If Π_1 and Π are

$$\frac{}{\Sigma; \Delta_1 \vdash t = t} \text{eq}\mathcal{R} \quad \frac{\left\{ \frac{\Pi_\theta}{\Sigma\theta; \Gamma\theta \vdash C\theta} \right\}_\theta}{\Sigma; t = t, \Gamma \vdash C} \text{eq}\mathcal{L}$$

then Ξ reduces to

$$\frac{\Sigma; \Delta_2 \vdash B_2 \quad \dots \quad \Sigma; \Delta_n \vdash B_n \quad \Sigma; \Delta_1, B_2, \dots, B_n, \Gamma \vdash C}{\Sigma; \Delta_1, \dots, \Delta_n, \Gamma \vdash C} mc$$

where ϵ is the empty substitution.

$\text{def}\mathcal{R}/\text{def}\mathcal{L}$: Suppose Π_1 and Π are

$$\frac{\Pi'_1}{\Delta_1 \vdash B[\vec{t}/\vec{x}]} \text{def}\mathcal{R} \quad \frac{\Pi'}{B[\vec{t}/\vec{x}], B_2, \dots, \Gamma \vdash C} \text{def}\mathcal{L}.$$

Then Ξ reduces to

$$\frac{\Pi'_1 \quad \Pi_2 \quad \dots \quad \Pi_n \quad B[\vec{t}/\vec{x}], \dots, \Gamma \vdash C}{\Delta_1, \dots, \Delta_n, \Gamma \vdash C} mc.$$

Left-commutative cases:

$\bullet\mathcal{L}/\circ\mathcal{L}$: Suppose Π ends with a left rule other than $c\mathcal{L}$ acting on B_1 and Π_1 is

$$\frac{\left\{ \frac{\Pi_1^i}{\Delta_1^i \vdash B_1} \right\}}{\Delta_1 \vdash B_1} \bullet\mathcal{L},$$

where $\bullet\mathcal{L}$ is any left rule except $\supset\mathcal{L}$, $\text{eq}\mathcal{L}$, or $\text{nat}\mathcal{L}$. Then Ξ reduces to

$$\frac{\left\{ \frac{\Pi_1^i}{\Delta_1^i \vdash B_1} \quad \left\{ \frac{\Pi_j}{\Delta_j \vdash B_j} \right\}_{j \in \{2..n\}} \quad B_1, \dots, B_n, \Gamma \vdash C}{\Delta_1^i, \Delta_2, \dots, \Delta_n, \Gamma \vdash C} mc \right\}}{\Delta_1, \Delta_2, \dots, \Delta_n, \Gamma \vdash C} \bullet\mathcal{L}.$$

$\supset \mathcal{L} / \circ \mathcal{L}$: Suppose Π ends with a left rule other than $c\mathcal{L}$ acting on B_1 and Π_1 is

$$\frac{\frac{\Pi'_1}{\Delta'_1 \vdash D'_1} \quad \frac{\Pi''_1}{D'_1, \Delta'_1 \vdash B_1}}{D'_1 \supset D'_1, \Delta'_1 \vdash B_1} \supset \mathcal{L} .$$

Let Ξ_1 be

$$\frac{\frac{\Pi''_1}{D'_1, \Delta'_1 \vdash B_1} \quad \frac{\Pi_2}{\Delta_2 \vdash B_2} \quad \dots \quad \frac{\Pi_n}{\Delta_n \vdash B_n} \quad B_1, \dots, B_n, \Gamma \vdash C}{D'_1, \Delta'_1, \Delta_2, \dots, \Delta_n, \Gamma \vdash C} mc .$$

Then Ξ reduces to

$$\frac{\frac{w(\Delta_2 \cup \dots \cup \Delta_n \cup \Gamma, \Pi'_1)}{\Delta'_1, \Delta_2, \dots, \Delta_n, \Gamma \vdash D'_1} \quad \frac{\Xi_1}{D'_1, \Delta'_1, \Delta_2, \dots, \Delta_n, \Gamma \vdash C}}{D'_1 \supset D'_1, \Delta'_1, \Delta_2, \dots, \Delta_n, \Gamma \vdash C} \supset \mathcal{L} .$$

$nat\mathcal{L} / \circ \mathcal{L}$: Suppose Π ends with a left rule other than $c\mathcal{L}$ acting on B_1 and Π_1 is

$$\frac{\frac{\Pi_1^1}{\vdash D_1 z} \quad \frac{\Pi_1^2}{D_1 j \vdash D_1(sj)} \quad \frac{\Pi_1^3}{D_1 I, \Delta'_1 \vdash B_1}}{nat I, \Delta'_1 \vdash B_1} nat\mathcal{L}$$

Let Ξ_1 be

$$\frac{\frac{\Pi_1^3}{D_1 I, \Delta'_1 \vdash B_1} \quad \frac{\Pi_2}{\Delta_2 \vdash B_2} \quad \dots \quad \frac{\Pi_n}{\Delta_n \vdash B_n} \quad B_1, \dots, B_n \vdash C}{D_1 I, \Delta'_1, \Delta_2, \dots, \Delta_n, \Gamma \vdash C} mc$$

Then Ξ reduces to

$$\frac{\frac{\Pi_1^1}{\vdash D_1 z} \quad \frac{\Pi_1^2}{D_1 j \vdash D_1(sj)} \quad \frac{\Xi_1}{D_1 I, \Delta'_1, \Delta_2, \dots, \Delta_n, \Gamma \vdash C}}{nat I, \Delta'_1, \Delta_2, \dots, \Delta_n, \Gamma \vdash C} nat\mathcal{L}$$

$eq\mathcal{L} / \circ \mathcal{L}$: If Π ends with a left rule other than $c\mathcal{L}$ acting on B_1 and Π_1 is

$$\frac{\left\{ \frac{\Pi^\theta}{\Delta'_1 \theta \vdash B_1 \theta} \right\}_\theta}{s = t, \Delta'_1 \vdash B_1} eq\mathcal{L}$$

then Ξ reduces to

$$\left\{ \frac{\frac{\Pi^\theta}{\Delta'_1 \theta \vdash B_1 \theta} \quad \frac{\Pi_2 \theta}{\Delta_2 \theta \vdash B_2 \theta} \quad \dots \quad \frac{\Pi_n \theta}{\Delta_n \theta \vdash B_n \theta} \quad B_1 \theta, \dots, B_n \theta, \Gamma \theta \vdash C \theta}{\Delta'_1 \theta, \Delta_2 \theta, \dots, \Delta_n \theta, \Gamma \theta \vdash C \theta} mc \right\}_{eq\mathcal{L}^\theta}$$

$$\frac{}{s = t, \Delta'_1, \Delta_2, \dots, \Delta_n, \Gamma \vdash C}$$

Right-commutative cases:

$- / \circ \mathcal{L}$: Suppose Π is

$$\frac{\left\{ \frac{\Pi^i}{B_1, \dots, B_n, \Gamma^i \vdash C} \right\}}{B_1, \dots, B_n, \Gamma \vdash C} \circ \mathcal{L} ,$$

where $\circ\mathcal{L}$ is any left rule other than $\supset\mathcal{L}$, $\text{eq}\mathcal{L}$, or $\text{nat}\mathcal{L}$ (but including $\text{c}\mathcal{L}$) acting on a formula other than B_1, \dots, B_n . The derivation Ξ reduces to

$$\frac{\left\{ \frac{\frac{\Pi_1}{\Delta_1 \vdash B_1} \dots \frac{\Pi_n}{\Delta_n \vdash B_n} \quad B_1, \dots, B_n, \Gamma^i \vdash C}{\Delta_1, \dots, \Delta_n, \Gamma^i \vdash C} mc \right\}}{\Delta_1, \dots, \Delta_n, \Gamma \vdash C} \circ\mathcal{L},$$

$-/\supset\mathcal{L}$: Suppose Π is

$$\frac{B_1, \dots, B_n, \Gamma' \vdash D' \quad B_1, \dots, B_n, D'', \Gamma' \vdash C}{B_1, \dots, B_n, D' \supset D'', \Gamma' \vdash C} \supset\mathcal{L}.$$

Let Ξ_1 be

$$\frac{\frac{\Pi_1}{\Delta_1 \vdash B_1} \dots \frac{\Pi_n}{\Delta_n \vdash B_n} \quad B_1, \dots, B_n, \Gamma' \vdash D'}{\Delta_1, \dots, \Delta_n, \Gamma' \vdash D'} mc$$

and Ξ_2 be

$$\frac{\frac{\Pi_1}{\Delta_1 \vdash B_1} \dots \frac{\Pi_n}{\Delta_n \vdash B_n} \quad B_1, \dots, B_n, D'', \Gamma' \vdash C}{\Delta_1, \dots, \Delta_n, D'', \Gamma' \vdash C} mc.$$

Then Ξ reduces to

$$\frac{\frac{\Xi_1}{\Delta_1, \dots, \Delta_n, \Gamma' \vdash D'} \quad \frac{\Xi_2}{\Delta_1, \dots, \Delta_n, D'', \Gamma' \vdash C}}{\Delta_1, \dots, \Delta_n, D' \supset D'', \Gamma' \vdash C} \supset\mathcal{L}.$$

$-/\text{nat}\mathcal{L}$: Suppose Π is

$$\frac{\frac{\Pi'}{\vdash Dz} \quad D j \vdash D(sj) \quad B_1, \dots, B_n, DI, \Gamma' \vdash C}{B_1, \dots, B_n, \text{nat } I, \Gamma' \vdash C} \text{nat}\mathcal{L}$$

Let Ξ_1 be

$$\frac{\frac{\Pi_1}{\Delta_1 \vdash B_1} \dots \frac{\Pi_n}{\Delta_n \vdash B_n} \quad B_1, \dots, B_n, DI, \Gamma' \vdash C}{\Delta_1, \dots, \Delta_n, DI, \Gamma' \vdash C} mc,$$

then Ξ reduces to

$$\frac{\frac{\Pi'}{\vdash Dz} \quad D j \vdash D(sj) \quad \frac{\Xi_1}{\Delta_1, \dots, \Delta_n, DI, \Gamma' \vdash C}}{\Delta_1, \dots, \Delta_n, \text{nat } I, \Gamma' \vdash C} \text{nat}\mathcal{L}$$

$-/\text{eq}\mathcal{L}$: If Π is

$$\frac{\left\{ B_1\rho, \dots, B_n\rho, \Gamma'\rho \vdash C\rho \right\}}{B_1, \dots, B_n, s = t, \Gamma' \vdash C} \text{eq}\mathcal{L},$$

then Ξ reduces to

$$\frac{\left\{ \frac{\left\{ \frac{\Pi_{i\rho}}{\Delta_{i\rho} \vdash B_{i\rho}} \right\}_{i \in \{1..n\}} \quad B_{i\rho}, \dots, \Gamma'\rho \vdash C\rho}{\Delta_{1\rho}, \dots, \Delta_{n\rho}, \Gamma'\rho \vdash C\rho} mc \right\}}{\Delta_1, \dots, \Delta_n, s = t, \Gamma' \vdash C} \text{eq}\mathcal{L}.$$

$-/\circ\mathcal{R}$: If Π is

$$\frac{\left\{ \frac{\Pi^i}{B_1, \dots, B_n, \Gamma^i \vdash C^i} \right\}}{B_1, \dots, B_n, \Gamma \vdash C} \circ\mathcal{R},$$

where $\circ\mathcal{R}$ is any right rule, then Ξ reduces to

$$\frac{\left\{ \frac{\frac{\Pi_1}{\Delta_1 \vdash B_1} \cdots \frac{\Pi'_n}{\Delta_n \vdash B_n} \frac{\Pi^i}{B_1, \dots, B_n, \Gamma^i \vdash C^i}}{\Delta_1, \dots, \Delta_n, \Gamma^i \vdash C^i} mc \right\}}{\Delta_1, \dots, \Delta_n, \Gamma \vdash C} \circ\mathcal{R}.$$

Multicut cases:

$mc/\circ\mathcal{L}$: If Π ends with a left rule other than $c\mathcal{L}$ acting on B_1 and Π_1 ends with a multicut and reduces to Π'_1 , then Ξ reduces to

$$\frac{\frac{\Pi'_1}{\Delta_1 \vdash B_1} \quad \frac{\Pi_2}{\Delta_2 \vdash B_2} \quad \cdots \quad \frac{\Pi_n}{\Delta_n \vdash B_n} \quad \frac{\Pi}{B_1, \dots, B_n, \Gamma \vdash C}}{\Delta_1, \dots, \Delta_n, \Gamma \vdash C} mc.$$

$-/mc$: Suppose Π is

$$\frac{\left\{ \frac{\Pi^j}{\{B_i\}_{i \in I^j}, \Gamma^j \vdash D^j} \right\}_{j \in \{1..m\}} \quad \frac{\Pi'}{\{D^j\}_{j \in \{1..m\}}, \{B_i\}_{i \in I'}, \Gamma' \vdash C}}{B_1, \dots, B_n, \Gamma^1, \dots, \Gamma^m, \Gamma' \vdash C} mc,$$

where I^1, \dots, I^m, I' partition the formulas $\{B_i\}_{i \in \{1..n\}}$ among the premise derivations $\Pi_1, \dots, \Pi_m, \Pi'$. For $1 \leq j \leq m$ let Ξ^j be

$$\frac{\left\{ \frac{\Pi_i}{\Delta_i \vdash B_i} \right\}_{i \in I^j} \quad \frac{\Pi^j}{\{B_i\}_{i \in I^j}, \Gamma^j \vdash D^j}}{\{\Delta_i\}_{i \in I^j}, \Gamma^j \vdash D^j} mc.$$

Then Ξ reduces to

$$\frac{\left\{ \frac{\Xi^j}{\dots \vdash D^j} \right\}_{j \in \{1..m\}} \quad \left\{ \frac{\Pi_i}{\Delta_i \vdash B_i} \right\}_{i \in I'} \quad \frac{\Pi'}{\dots \vdash C}}{\Delta_1, \dots, \Delta_n, \Gamma^1, \dots, \Gamma^m, \Gamma' \vdash C} mc.$$

Structural case:

$-/c\mathcal{L}$: If Π is

$$\frac{\frac{\Pi'}{B_1, B_1, B_2, \dots, B_n, \Gamma \vdash C}}{B_1, B_2, \dots, B_n, \Gamma \vdash C} c\mathcal{L},$$

then Ξ reduces to

$$\frac{\frac{\frac{\Pi_1}{\Delta_1 \vdash B_1} \quad \left\{ \frac{\Pi_i}{\Delta_i \vdash B_i} \right\}_{i \in \{1..n\}} \quad \frac{\Pi'}{B_1, B_1, B_2, \dots, B_n, \Gamma \vdash C}}{\Delta_1, \Delta_1, \Delta_2, \dots, \Delta_n, \Delta_n, \Gamma \vdash C} mc}{\Delta_1, \Delta_2, \dots, \Delta_n, \Gamma \vdash C} c\mathcal{L}.$$

Axiom cases:

$id_\pi / \circ \mathcal{L}$: Suppose Π ends with either $nat\mathcal{L}$ or $eq\mathcal{L}$ on B_1 and Π_1 ends with the id_π rule:

$$\frac{\pi_1.B = \pi_2.B_1}{\Delta'_1, B \vdash B_1} id_\pi$$

Then it is the case that $B = \pi_1^{-1}.\pi_2.B_1$. Apply the construction in Definition 8 to Π to get a derivation Π' of $B, B_2, \dots, B_n, \Gamma \vdash C$. The derivation Ξ reduces to

$$\frac{\frac{\Pi_2}{\Delta_2 \vdash B_2} \cdots \frac{\Pi_n}{\Delta_n \vdash B_n} \quad \frac{w(\Delta'_1, \Pi')}{B, \Delta'_1, B_2, \dots, B_n, \Gamma \vdash C}}{B, \Delta'_1, \Delta_2, \dots, \Delta_n, \Gamma \vdash C} mc$$

$-/id_\pi$: If Π ends with the id_π rule with a matching formula in Γ , i.e., there exists $C' \in \Gamma$ such that $\pi.C' = \pi'.C$ for some permutations π and π' , then then Ξ reduces to

$$\frac{}{\Delta_1, \dots, \Delta_n, \Gamma \vdash C} id_\pi$$

If Π ends with the id_π rule but C does not match any formula in Γ , then C must match one of the cut formulas, say B_1 , i.e., there exists permutations π_1 and π_2 such that $\pi_1.B_1 = \pi_2.C$. That is, $C = \pi_2^{-1}.\pi_1.B_1$. In this case, we first apply the permutation $\pi_2^{-1}.\pi_1$ to Π_1 according to the construction in Definition 8 to get a derivation Π'_1 of $\Delta_1 \vdash \pi_2^{-1}.\pi_1.B_1$. Ξ then reduces to $w(\Delta_2 \cup \dots \cup \Delta_n \cup \Gamma, \Pi'_1)$. \square

An inspection of the rules of the logic and this definition will reveal that every derivation ending with a multicut has a reduct. Because we use a multiset as the left side of the sequent, there may be ambiguity as to whether a formula occurring on the left side of the rightmost premise to a multicut rule is in fact a cut formula, and if so, which of the left premises corresponds to it. As a result, several of the reduction rules may apply, and so a derivation may have multiple reducts.

5 Normalizability and reducibility

We now define two properties of derivations: normalizability and reducibility. Each of these properties implies that the derivation can be reduced to a cut-free derivation of the same end-sequent. In the following, substitutions mean Σ -substitutions for some signature Σ . The definitions are similar to those by McDowell and Miller [3]. However, since the cut reduction in our case involves several transformations of derivations, other than substitutions and weakening, we need to build this transformations into the definitions of normalizability and reducibility.

Definition 19. A height-preserving (HP) transformation \mathcal{T} is a finite sequence of transformations $\mathcal{F}_1, \dots, \mathcal{F}_n$ where each \mathcal{F}_i is one of the transformations described in Definition 6, Definition 7, Definition 8 and Definition 9. The number n is the order of \mathcal{T} . The application of \mathcal{T} to Π is defined as follows:

$$\begin{aligned} \mathcal{T}_0(\Pi) &= \Pi \\ \mathcal{T}_{i+1}(\Pi) &= \mathcal{F}_{i+1}(\mathcal{T}_i(\Pi)) \\ \mathcal{T}(\Pi) &= \mathcal{T}_n(\Pi) \end{aligned}$$

Note that a height-preserving transformation may not be defined for all derivations, and that it may be the identity transformation (i.e., it does nothing). Height-preserving transformations are ranged over by $\mathcal{T}, \mathcal{F}, \mathcal{G}$ and \mathcal{H} .

Lemma 20. Let \mathcal{T} be a height-preserving transformation. For any derivation Π , if $\mathcal{T}(\Pi)$ is defined, then $ht(\mathcal{T}(\Pi)) \leq ht(\Pi)$.

Definition 21. We define the set of normalizable derivations to be the smallest set that satisfies the following conditions:

1. If a derivation Π ends with a multicut, then it is normalizable if for every height-preserving transformation \mathcal{T} such that $\mathcal{T}(\Pi)$ is defined, there is a normalizable reduct of $\mathcal{T}(\Pi)$.
2. If a derivation ends with any rule other than a multicut, then it is normalizable if the premise derivations are normalizable.

These clauses assert that a given derivation is normalizable provided certain (perhaps infinitely many) other derivations are normalizable. If we call these other derivations the predecessors of the given derivation, then a derivation is normalizable if and only if the tree of the derivation and its successive predecessors is well-founded. In this case, the well-founded tree is called the normalization of the derivation.

The set of normalizable derivations is not empty; the cut-free proofs, for instance, are normalizable.

Since a normalization is well-founded, it has an associated induction principle: for any property P of derivations, if for every derivation Π in the normalization, P holds for every predecessor of Π implies that P holds for Π , then P holds for every derivation in the normalization.

Lemma 22. If there is a normalizable derivation of a sequent, then there is a cut-free derivation of the sequent.

Proof. Let Π be a normalizable derivation of the sequent $\Gamma \vdash B$. We show by induction on the normalization of Π that there is a cut-free derivation of $\Gamma \vdash B$.

1. If Π ends with a multicut, then any of its reducts is one of its predecessors and so is normalizable. One of its reduct, via the empty transformation, is also a derivation of $\Gamma \vdash B$, so by the induction hypothesis this sequent has a cut-free derivation.
2. Suppose Π ends with a rule other than multicut. Since we are given that Π is normalizable, by definition the premise derivations are normalizable. These premise derivations are the predecessors of Π , so by the induction hypothesis there are cut-free derivations of the premises. Thus there is a cut-free derivation of $\Gamma \vdash B$.

□

The next four lemmas are also proved by induction on the normalization of derivations.

Lemma 23. If Π is a normalizable derivation, then for any substitution θ such that $\Pi\theta$ is defined, $\Pi\theta$ is normalizable.

Lemma 24. If Π is normalizable, then for any multiset of formulas Δ , if $w(\Delta, \Pi)$ is defined, then $w(\Delta, \Pi)$ is normalizable.

Lemma 25. If Π is normalizable, then for any permutations $\bar{\pi}$ such that $\langle \bar{\pi} \rangle.\Pi$ is defined, $\langle \bar{\pi} \rangle.\Pi$ is normalizable.

Lemma 26. If Π is normalizable, then for any nominal constants \vec{a} such that $r(x, \vec{a}, \Pi)$ is defined, $r(x, \vec{a}, \Pi)$ is normalizable.

Lemma 27. If Π is normalizable, then for any height-preserving transformation \mathcal{T} such that $\mathcal{T}(\Pi)$ is defined, $\mathcal{T}(\Pi)$ is normalizable.

Definition 28. The level of a sequent $\Gamma \vdash C$ is the level of C . The level of a derivation Π is the level of its root sequent.

The definition of reducibility for derivations is done by induction on the level of derivations: in defining the reducibility of level- i derivations, we assume that the reducibility of derivations of level j , for all $j < i$ is already defined. In the following definition, when we apply a transformation \mathcal{T} to a derivation Π of $B_1, \dots, B_n \vdash B_0$, we use the notation $\mathcal{T}(B_i)$ to denote the formula in the root sequent of $\mathcal{T}(\Pi)$ that results from applying the transformation to B_i .

Definition 29. Reducibility. For any i , we define the set of reducible i -level derivations to be the smallest set of i -level derivations that satisfies the following conditions:

1. If a derivation Π ends with a multicut then it is reducible if for every height-preserving transformation \mathcal{T} such that $\mathcal{T}(\Pi)$ is defined, there is a reducible reduct of $\mathcal{T}(\Pi)$.
2. Suppose the derivation ends with the implication right rule

$$\frac{B, \Gamma \vdash^{\Pi} C}{\Gamma \vdash B \supset C} \supset \mathcal{R}$$

Then the derivation is reducible if Π is reducible and for every height-preserving transformation \mathcal{T} such that $\mathcal{T}(\Pi)$ is defined, multiset of formulas Δ and reducible derivation Π' of $\Delta \vdash B'$, where $B' = \mathcal{T}(B)$, the derivation

$$\frac{\Delta \vdash^{ \Pi' } B' \quad B', \Gamma' \vdash^{ \mathcal{T}(\Pi) } C'}{\Delta, \Gamma' \vdash C'} mc$$

is reducible.

3. If the derivation ends with the implication left rule or the nat rule, then it is reducible if the right premise derivation is reducible and the other premise derivations are normalizable.
4. If the derivation ends with any other rule, then it is reducible if the premise derivations are reducible.

These clauses assert that a given derivation is reducible provided certain other derivations are reducible. If we call these other derivations the predecessors of the given derivation, then a derivation is reducible only if the tree of the derivation and its successive predecessors is well founded. In this case, the well founded tree is called the reduction of the derivation.

Lemma 30. If a derivation is reducible, then it is normalizable.

Proof. By induction on the reduction of the derivation. □

Lemma 31. If a derivation Π is reducible, then for any height-preserving \mathcal{T} such that $\mathcal{T}(\Pi)$ is defined, $\mathcal{T}(\Pi)$ is reducible.

Proof. By induction on the reduction of Π and Lemma 27.

6 Cut elimination

In the following, when we mention $\mathcal{T}(\Pi)$ we assume implicitly that it is defined. We shall also use the notation $\underline{B}_{\mathcal{T}}$ to denote $\mathcal{T}(B)$, that is the application of the transformation to the formula B . Similarly, the multiset $\mathcal{T}(\Delta)$ will be written $\underline{\Delta}_{\mathcal{T}}$. We drop the subscript \mathcal{T} if it is clear from context which transformation we refer to.

Lemma 32. For any derivation Π of $\Sigma; B_1, \dots, B_n, \Gamma \vdash C$ and reducible derivations Π_1, \dots, Π_n of $\Sigma; \Delta_1 \vdash C_1, \dots, \Sigma; \Delta_n \vdash C_n$, where $n \geq 0$, and for any transformations $\mathcal{T}_1, \dots, \mathcal{T}_n, \mathcal{T}$ such that $\mathcal{T}_i(\Pi_i)$ is defined and $\mathcal{T}_i(C_i) = \mathcal{T}(B_i)$, the derivation Ξ

$$\frac{\Sigma'; \underline{\Delta}_{\mathcal{T}_1} \vdash \underline{B}_{\mathcal{T}_1} \quad \dots \quad \Sigma'; \underline{\Delta}_{\mathcal{T}_n} \vdash \underline{B}_{\mathcal{T}_n} \quad \Sigma'; \underline{B}_{\mathcal{T}_1}, \dots, \underline{B}_{\mathcal{T}_n}, \underline{\Gamma}_{\mathcal{T}} \vdash \underline{C}_{\mathcal{T}}}{\Sigma'; \underline{\Delta}_{\mathcal{T}_1}, \dots, \underline{\Delta}_{\mathcal{T}_n}, \underline{\Gamma}_{\mathcal{T}} \vdash \underline{C}_{\mathcal{T}}} mc$$

is reducible.

Proof. The proof is by induction on $ht(\Pi)$ with subordinate induction on n and on the reductions of Π_1, \dots, Π_n . Since the proof does not depend on the order of the inductions on reductions, when we need to distinguish of one the Π_i 's we shall refer to it as Π_1 without loss of generality.

We need to show that for every \mathcal{T}' , the derivation every reduct of $\mathcal{T}'(\Xi)$ is reducible. If $n = 0$ then $\mathcal{T}'(\Xi)$ reduces to $\mathcal{T}'(\mathcal{T}(\Pi))$. Since reducibility is preserved by height-preserving transformation, it suffices to consider the case where \mathcal{T} and \mathcal{T}' are the identity transformation, that is, we need only to show that Π is reducible. This is proved by case analysis on the last rule of Π . For each case, the results follow from the outer induction hypothesis and Definition 29. The case with $\supset \mathcal{R}$ requires that height-preserving transformations do not increase the height of the derivations (see Lemma 20). In the cases for $\supset \mathcal{L}$ and $nat\mathcal{L}$ we need the additional information that reducibility implies normalizability (see Lemma 30).

For $n > 0$, we analyze all possible reductions that apply to $\mathcal{T}'(\Xi)$ and show that every reduct of $\mathcal{T}'(\Xi)$ is reducible. We suppose that $\mathcal{T}'(\Xi)$ is of the following form:

$$\frac{\frac{\mathcal{F}_1(\Pi_1)}{\underline{\Delta}_1 \vdash \underline{C}_1}_{\mathcal{F}_1} \quad \dots \quad \frac{\mathcal{F}_n(\Pi_n)}{\underline{\Delta}_n \vdash \underline{C}_n}_{\mathcal{F}_n} \quad \frac{\mathcal{F}(\Pi)}{\underline{B}_1, \dots, \underline{B}_n, \underline{\Gamma} \vdash \underline{C}}_{\mathcal{F}}}{\underline{\Delta}_1, \dots, \underline{\Delta}_n, \underline{\Gamma} \vdash \underline{C}}_{mc} \quad mc$$

where $\underline{B}_i = \underline{C}_{i\mathcal{F}_i}$. In several cases below, we often omit the subscripts \mathcal{F} or \mathcal{F}_i when it is clear from context which transformations we refer to. We also often switch between $\underline{B}_{i\mathcal{F}}$ and $\underline{C}_{i\mathcal{F}_i}$ to make the inference figures more readable.

Most cases follow immediately from the inductive hypothesis and Definition 29 and Lemma 30, Lemma 31 and Lemma 20. We show here the interesting cases.

$\supset \mathcal{R} / \supset \mathcal{L}$: Suppose Π_1 and Π are

$$\frac{\frac{\Pi'_1}{\underline{\Delta}_1, \underline{B}'_1 \vdash \underline{B}''_1}}{\underline{\Delta}_1 \vdash \underline{B}'_1 \supset \underline{B}''_1} \supset \mathcal{R} \quad \frac{\frac{\Pi'}{B_2, \dots, \Gamma \vdash B'_1} \quad \frac{\Pi''}{B'_1, B_2, \dots, \Gamma \vdash C}}{B'_1 \supset B''_1, B_2, \dots, B_n, \Gamma \vdash C} \supset \mathcal{L}.$$

Let Ξ_1 be the derivation

$$\frac{\frac{\mathcal{F}_2(\Pi_2)}{\underline{\Delta}_2 \vdash \underline{B}_2} \quad \dots \quad \frac{\mathcal{F}_n(\Pi_n)}{\underline{\Delta}_n \vdash \underline{B}_n} \quad \frac{\mathcal{F}_n(\Pi')}{\underline{B}_2, \dots, \underline{B}_n, \underline{\Gamma} \vdash \underline{B}'_1}}{\underline{\Delta}_2, \dots, \underline{\Delta}_n, \underline{\Gamma} \vdash \underline{B}'_1}_{mc}$$

Then Ξ_1 is reducible by induction hypothesis since \mathcal{F} and \mathcal{F}_i preserve reducibility (Lemma 31) and do not increase the height of derivations (Lemma 20). Since we are given that Π_1 is reducible, by Definition 29, the derivation Ξ_2

$$\frac{\frac{\Xi_1}{\underline{\Delta}_2, \dots, \underline{\Delta}_n, \underline{\Gamma} \vdash \underline{B}'_1} \quad \frac{\mathcal{F}_1(\Pi'_1)}{\underline{B}'_1, \underline{\Delta}_1 \vdash \underline{B}''_1}}{\underline{\Delta}_1, \dots, \underline{\Delta}_n, \underline{\Gamma} \vdash \underline{B}''_1}_{mc}$$

is reducible as well. Therefore, the reduct of $\mathcal{T}'(\Xi)$

$$\frac{\frac{\Xi_2}{\dots \vdash \underline{B}''_1} \quad \frac{\left\{ \frac{\mathcal{F}_i(\Pi_i)}{\underline{\Delta}_i \vdash \underline{B}_i} \right\}_{i \in \{2..n\}} \quad \frac{\mathcal{F}(\Pi'')}{\underline{B}''_1, \{ \underline{B}_i \}_{i \in \{2..n\}}, \underline{\Gamma} \vdash \underline{C}}}{\underline{\Delta}_1, \dots, \underline{\Delta}_n, \underline{\Gamma}, \underline{\Delta}_2, \dots, \underline{\Delta}_n, \underline{\Gamma} \vdash \underline{C}}_{mc} \quad c\mathcal{L} \quad \underline{\Delta}_1, \dots, \underline{\Delta}_n, \underline{\Gamma} \vdash \underline{C}.$$

is reducible by the outer induction hypothesis and Definition 29.

$\forall\mathcal{R}/\forall\mathcal{L}$: Suppose Π_1 and Π are

$$\frac{\frac{\Pi'_1}{\Sigma, h; \Delta_1 \vdash B[h \vec{c}/x]} \forall\mathcal{R}}{\Sigma; \Delta_1 \vdash \forall x.B} \quad \frac{\frac{\Pi'}{\Sigma; B[t/x], B_2, \dots, B_n, \Gamma \vdash C} \forall\mathcal{L}}{\Sigma; \forall x.B, B_2, \dots, B_n, \Gamma \vdash C} \forall\mathcal{L}$$

Applying the transformation \mathcal{F}_1 to Π_1 (and similarly, \mathcal{F} to Π) might require several transformation be done on the premise of the derivation, e.g., to avoid clashes of nominal constants, etc., so let us suppose that $\mathcal{F}_1(\Pi_1)$ and $\mathcal{F}(\Pi)$ are of the following shapes:

$$\frac{\frac{\mathcal{G}_1(\Pi'_1)}{\Sigma', h; \Delta_1 \vdash D[h' \vec{d}/x]} \forall\mathcal{R}}{\Sigma'; \underline{\Delta_1} \vdash \forall x.D} \quad \frac{\frac{\mathcal{G}(\Pi')}{\Sigma'; D[s/x], \underline{B_2}, \dots, \underline{B_n}, \underline{\Gamma} \vdash \underline{C}} \forall\mathcal{L}}{\Sigma'; \forall x.D, \underline{B_2}, \dots, \underline{B_n}, \underline{\Gamma} \vdash \underline{C}} \forall\mathcal{L}$$

where $\forall x.D = \forall x.B$ and $D[s/x] = B[t/x]$. If the support of $D[s/x]$ is larger than $\{\vec{d}\}$, then the reduction rule for $\forall\mathcal{R}/\forall\mathcal{L}$ requires further transformations be applied to $\mathcal{G}_1(\Pi'_1)$, i.e., as is described in Lemma 17. So let us suppose that this transformation is applied, resulting in a derivation

$$\frac{\mathcal{G}'_1(\Pi'_1)}{\Sigma', f; \underline{\Delta_1} \vdash D[f \vec{e}/x]} .$$

Then $\mathcal{T}'(\Xi)$ reduces to

$$\frac{\frac{\mathcal{G}'_1(\Pi'_1)[\lambda \vec{e}.s/f]}{\Sigma'; \underline{\Delta_1} \vdash D[s/x]} \quad \frac{\mathcal{F}_2(\Pi_2)}{\underline{\Delta_2} \vdash \underline{B_2}} \quad \dots \quad \frac{\mathcal{F}_n(\Pi_n)}{\underline{\Delta_n} \vdash \underline{B_n}} \quad \frac{\mathcal{G}(\Pi')}{\Sigma'; D[s/x], \dots, \underline{\Gamma} \vdash \underline{C}}}{\Sigma'; \underline{\Delta_1}, \dots, \underline{\Delta_n}, \underline{\Gamma} \vdash \underline{C}} mc$$

which is reducible by the outer induction hypothesis.

$\text{nat}\mathcal{R}/\text{nat}\mathcal{L}$: Suppose Π_1 and Π are

$$\frac{\frac{\Pi'_1}{\Delta_1 \vdash \text{nat } M}}{\Delta_1 \vdash \text{nat } M} \text{nat}\mathcal{R} \quad \frac{\frac{\Pi'}{\vdash D z} \quad \frac{\Pi''}{D j \vdash D(s j)} \quad \frac{\Pi'''}{D(s M), B_2, \dots, B_n, \Gamma \vdash C}}{\text{nat}(s I), B_2, \dots, B_n, \Gamma \vdash C} \text{nat}\mathcal{L}$$

then $\mathcal{F}_1(\Pi_1)$ and $\mathcal{F}(\Pi)$ are

$$\frac{\frac{\mathcal{F}_1(\Pi'_1)}{\underline{\Delta_1} \vdash \text{nat } I}}{\underline{\Delta_1} \vdash \text{nat } I} \text{nat}\mathcal{R} \quad \frac{\frac{\Pi'}{\vdash D z} \quad \frac{\Pi''}{D j \vdash D(s j)} \quad \frac{\mathcal{F}(\Pi''')}{D(s I), \underline{B_2}, \dots, \underline{B_n}, \underline{\Gamma} \vdash \underline{C}}}{\text{nat}(s I), \underline{B_2}, \dots, \underline{B_n}, \underline{\Gamma} \vdash \underline{C}} \text{nat}\mathcal{L}$$

Note that the derivations Π' and Π'' are not affected by the transformation \mathcal{F} since D is a closed term with no occurrences of nominal constants and j in Π'' is a new eigenvariable. Let Ξ_1 be the derivation

$$\frac{\frac{\mathcal{F}_1(\Pi'_1)}{\underline{\Delta_1} \vdash \text{nat } I} \quad \frac{\frac{\Pi'}{\vdash D z} \quad \frac{\Pi''}{D j \vdash D(s j)} \quad \frac{\overline{D I \vdash D I}}{\text{nat } I \vdash D I} id_\pi}{\underline{\Delta_1} \vdash D I} \text{nat}\mathcal{L} \quad mc .$$

Since the height of the right premise is no larger than $ht(\Pi)$, and Π'_1 is a predecessor of Π_1 , Ξ_1 is reducible by induction on the reduction of Π_1 . Let $\{\vec{c}\}$ be the support of I . We construct the derivation Π^\bullet of $h; D(h \vec{c}) \vdash D(s(h \vec{c}))$ from Π'' using the procedures described in Definition 7 and Definition 9. Let Ξ_2 be

$$\frac{\frac{\Xi_1}{\underline{\Delta_1} \vdash D I} \quad \frac{\Pi^\bullet[\lambda \vec{c}.I/h]}{D I \vdash D(s I)}}{\underline{\Delta_1} \vdash D(s I)} mc.$$

Since $ht(\Pi^\bullet[\lambda\vec{c}.I/h]) \leq ht(\Pi'')$, by the outer induction hypothesis, Ξ_2 is also reducible. Therefore the reduct of $\mathcal{T}'(\Xi)$

$$\frac{\frac{\Xi_2}{\underline{\Delta}_1 \vdash D(sI)} \quad \frac{\mathcal{F}_2(\Pi_2)}{\underline{\Delta}_2 \vdash \underline{B}_2} \quad \dots \quad \frac{\mathcal{F}_n(\Pi_n)}{\underline{\Delta}_n \vdash \underline{B}_n} \quad \frac{\mathcal{F}(\Pi''')}{D(sI), \underline{B}_2, \dots, \underline{\Gamma} \vdash \underline{C}}}{\underline{\Delta}_1, \dots, \underline{\Delta}_2, \underline{\Gamma} \vdash \underline{C}} mc$$

is reducible by the outer induction hypothesis.

$\text{eq}\mathcal{L}/\circ\mathcal{L}$: Suppose Π_1 is

$$\frac{\left\{ \frac{\Pi^\theta}{\underline{\Delta}_1 \theta \vdash B_1 \theta} \right\}_\theta}{s = t, \underline{\Delta}_1 \vdash B_1} \text{eq}\mathcal{L}$$

then $\mathcal{F}_1(\Pi_1)$ is

$$\frac{\left\{ \frac{\Pi^{\bullet\rho}}{\underline{\Delta}_1 \theta \vdash \underline{B}_1 \theta} \right\}_\rho}{\underline{s} = \underline{t}, \underline{\Delta}_1 \vdash \underline{B}_1} \text{eq}\mathcal{L}$$

where each $\Pi^{\bullet\rho}$ is obtained from some Π^θ by the transformations described in Definition 6, Definition 7, Definition 8 and Definition 9. We denote with $f(\rho)$ the substitution θ such that $\Pi^{\bullet\rho}$ is constructed out of Π^θ . Thus we can write each $\Pi^{\bullet\rho}$ as the derivation $\mathcal{F}_\rho(\Pi^{f(\rho)})$ for some transformation \mathcal{F}_ρ . The reduct of $\mathcal{T}'(\Xi)$

$$\frac{\left\{ \frac{\mathcal{F}_\rho(\Pi^{f(\rho)})}{\underline{\Delta}'_1 \rho \vdash \underline{B}_1 \rho} \quad \frac{\mathcal{F}_2(\Pi_2)\rho}{\underline{\Delta}_2 \rho \vdash \underline{B}_2 \rho} \quad \dots \quad \frac{\mathcal{F}_n(\Pi_n)\rho}{\underline{\Delta}_n \rho \vdash \underline{B}_n \rho} \quad \frac{\mathcal{F}(\Pi)\rho}{\underline{B}_1 \rho, \dots, \underline{B}_n \rho, \underline{\Gamma} \rho \vdash \underline{C} \rho} mc \right\}}{\underline{s} = \underline{t}, \underline{\Delta}'_1, \underline{\Delta}_2, \dots, \underline{\Delta}_n, \underline{\Gamma} \vdash \underline{C}}^{\rho} \text{eq}\mathcal{L}$$

Each premise derivation of the above derivation is reducible by the induction hypothesis on the reduction of Π_1 , since each $\Pi^{f(\rho)}$ is a predecessor of Π_1 . The reduct of $\mathcal{T}'(\Xi)$ is therefore reducible by Definition 29.

$-/\supset\mathcal{R}$: Suppose Π is

$$\frac{\frac{\mathcal{F}(\Pi')}{B_1, \dots, B_n, \underline{\Gamma}, C_1 \vdash C_2}}{B_1, \dots, B_n, \underline{\Gamma} \vdash C_1 \supset C_2} \supset\mathcal{R}$$

then $\mathcal{F}_1(\Pi)$

$$\frac{\frac{\mathcal{F}(\Pi')}{\underline{B}_1, \dots, \underline{B}_n, \underline{\Gamma}, \underline{C}_1 \vdash \underline{C}_2}}{\underline{B}_1, \dots, \underline{B}_n, \underline{\Gamma} \vdash \underline{C}_1 \supset \underline{C}_2} \supset\mathcal{R}$$

Let Ξ_1 be

$$\frac{\frac{\mathcal{F}_1(\Pi_1)}{\underline{\Delta}_1 \vdash \underline{B}_1} \quad \dots \quad \frac{\mathcal{F}_n(\Pi_n)}{\underline{\Delta}_n \vdash \underline{B}_n} \quad \frac{\mathcal{F}(\Pi')}{\underline{B}_1, \dots, \underline{B}_n, \underline{\Gamma}, \underline{C}_1 \vdash \underline{C}_2}}{\underline{\Delta}_1, \dots, \underline{\Delta}_n, \underline{C}_1 \vdash \underline{C}_2}$$

which is reducible by the outer induction hypothesis. Let Ξ_2 be the derivation

$$\frac{\frac{\Xi_1}{\underline{\Delta}_1, \dots, \underline{\Delta}_n, \underline{\Gamma}, \underline{C}_1 \vdash \underline{C}_2}}{\underline{\Delta}_1, \dots, \underline{\Delta}_n, \underline{\Gamma} \vdash \underline{C}_1 \supset \underline{C}_2} \supset\mathcal{R} ,$$

which is the reduct of $\mathcal{T}'(\Xi)$. To show that Ξ_2 is reducible, we need to show that for any \mathcal{T}'' , and for any derivation Π'' of $\Delta \vdash D$, where $D = \mathcal{T}''(\underline{C_1})$, the derivation Ξ_3

$$\frac{\frac{\Pi''}{\Delta \vdash D} \quad D, \underline{\Delta_{1\mathcal{G}_1}}, \dots, \underline{\Delta_{n\mathcal{G}_n}}, \underline{\Gamma_{\mathcal{G}}} \vdash \underline{C_{2\mathcal{G}}}}{\Delta, \underline{\Delta_{1\mathcal{G}_1}}, \dots, \underline{\Delta_{n\mathcal{G}_n}}, \underline{\Gamma_{\mathcal{G}}} \vdash \underline{C_{2\mathcal{G}}}} mc$$

is reducible. Here the transformations \mathcal{G}_i and \mathcal{G} are transformations associated with the premise derivations in $\mathcal{T}''(\Xi_2)$. Ξ_3 is reducible if for any transformation \mathcal{H} , every reduct of the derivation $\mathcal{H}(\Xi_3)$ is reducible. The reduct of $\mathcal{H}(\Xi_3)$ in this case is:

$$\frac{\frac{\mathcal{H}'(\Pi'')}{\underline{\Delta} \vdash \underline{D}} \quad \frac{\mathcal{H}_1(\Pi_1)}{\underline{\Delta_1} \vdash \underline{B_1}} \quad \dots \quad \frac{\mathcal{H}_n(\Pi_n)}{\underline{\Delta_n} \vdash \underline{B_n}} \quad \underline{D}, \underline{B_1}, \dots, \underline{B_n}, \underline{\Gamma} \vdash \underline{C_2}}{\underline{\Delta}, \underline{\Delta_1}, \dots, \underline{\Delta_n}, \underline{\Gamma} \vdash \underline{C_2}} mc$$

where $\mathcal{H}_1, \dots, \mathcal{H}_n$ and \mathcal{H}'' are transformations applied to the premises of $\mathcal{H}(\mathcal{T}''(\Xi_2))$ and \mathcal{H}' is the transformation applied to the left premise of $\mathcal{H}(\Xi_3)$. This derivation is reducible by the outer induction hypothesis. \square

Corollary 33. *Every derivation is reducible.*

Proof. This result follows immediately from Lemma 32 with $n = 0$. \square

Theorem 34. *The cut rule is admissible in LG^ω .*

Proof. Follows immediately from Corollary 33, Lemma 30 and Lemma 22. \square

Corollary 35. *The logic LG^ω is consistent, i.e., it is not the case that both A and $A \supset \perp$ are provable.*

7 Correspondence between LG and $FO\lambda^\nabla$

We now show that the formulation of LG is equivalent to $FO\lambda^\nabla$ extended with the axiom schemes of name permutations and weakening:

$$\nabla x \nabla y. B x y \supset \nabla y \nabla x. B x y \quad \text{and} \quad B \equiv \nabla x. B \quad (2)$$

where x is not free in B in the second scheme.

Sequents in $FO\lambda^\nabla$ are expressions of the form

$$\Sigma; \sigma_1 \triangleright B_1, \dots, \sigma_n \triangleright B_n \vdash \sigma_0 \triangleright B_0.$$

Σ is the *signature* of the sequent, σ_i is a list of variables locally scoped over B_i , and is referred to as *local signature*. The expression $\sigma_i \triangleright B_i$ is called a *local judgment*, or *judgment* for short. In [8], local judgments are considered equal modulo renaming of their local signatures, e.g., $(a, b) \triangleright P a b$ is equal to $(c, d) \triangleright P c d$. Local judgments are ranged over by scripted capital letters, e.g., \mathcal{B} , \mathcal{D} , etc. For the purpose of proving the correspondence with LG , however, we will make this renaming step explicit, by including the rules:

$$\frac{\vec{y} \triangleright B', \Gamma \vdash C}{\vec{x} \triangleright B, \Gamma \vdash C} \alpha_{\mathcal{R}}, \lambda \vec{x}. B \equiv_{\alpha} \lambda \vec{y}. B' \quad \frac{\Gamma \vdash \vec{y} \triangleright B'}{\Gamma \vdash \vec{x} \triangleright B} \alpha_{\mathcal{L}}, \lambda \vec{x}. B \equiv_{\alpha} \lambda \vec{y}. B'$$

The inference rules of $FO\lambda^\nabla$ are given in Figure 2.

We now consider the correspondence between LG with $FO\lambda^\nabla$ extended with the following axiom schemes:

$$\nabla x \nabla y. B x y \equiv \nabla y \nabla x. B x y. \quad (3)$$

$$\begin{array}{c}
\frac{}{\Sigma; \sigma \triangleright B, \Gamma \vdash \sigma \triangleright B} id \quad \frac{\Sigma; \Delta \vdash B \quad \Sigma; B, \Gamma \vdash C}{\Sigma; \Delta, \Gamma \vdash C} cut \\
\frac{\Sigma; \sigma \triangleright B, \sigma \triangleright C, \Gamma \vdash \mathcal{D}}{\Sigma; \sigma \triangleright B \wedge C, \Gamma \vdash \mathcal{D}} \wedge \mathcal{L} \quad \frac{\Sigma; \Gamma \vdash \sigma \triangleright B \quad \Sigma; \Gamma \vdash \sigma \triangleright C}{\Sigma; \Gamma \vdash \sigma \triangleright B \wedge C} \wedge \mathcal{R} \\
\frac{\Sigma; \sigma \triangleright B, \Gamma \vdash \mathcal{D} \quad \Sigma; \sigma \triangleright C, \Gamma \vdash \mathcal{D}}{\Sigma; \sigma \triangleright B \vee C, \Gamma \vdash \mathcal{D}} \vee \mathcal{L} \quad \frac{\Sigma; \Gamma \vdash \sigma \triangleright B}{\Sigma; \Gamma \vdash \sigma \triangleright B \vee C} \vee \mathcal{R} \\
\frac{}{\Sigma; \sigma \triangleright \perp, \Gamma \vdash B} \perp \mathcal{L} \quad \frac{\Sigma; \Gamma \vdash \sigma \triangleright C}{\Sigma; \Gamma \vdash \sigma \triangleright B \vee C} \vee \mathcal{R} \\
\frac{\Sigma; \Gamma \vdash \sigma \triangleright B \quad \Sigma; \sigma \triangleright C, \Gamma \vdash \mathcal{D}}{\Sigma; \sigma \triangleright B \supset C, \Gamma \vdash \mathcal{D}} \supset \mathcal{L} \quad \frac{\Sigma; \sigma \triangleright B, \Gamma \vdash \sigma \triangleright C}{\Sigma; \Gamma \vdash \sigma \triangleright B \supset C} \supset \mathcal{R} \\
\frac{\Sigma, \sigma \vdash t : \gamma \quad \Sigma; \sigma \triangleright B[t/x], \Gamma \vdash C}{\Sigma; \sigma \triangleright \forall_\gamma x. B, \Gamma \vdash C} \forall \mathcal{L} \quad \frac{\Sigma, h; \Gamma \vdash \sigma \triangleright B[(h \sigma)/x]}{\Sigma; \Gamma \vdash \sigma \triangleright \forall x. B} \forall \mathcal{R} \\
\frac{\Sigma, h; \sigma \triangleright B[(h \sigma)/x], \Gamma \vdash C}{\Sigma; \sigma \triangleright \exists x. B, \Gamma \vdash C} \exists \mathcal{L} \quad \frac{\Sigma, \sigma \vdash t : \gamma \quad \Sigma; \Gamma \vdash \sigma \triangleright B[t/x]}{\Sigma; \Gamma \vdash \sigma \triangleright \exists_\gamma x. B} \exists \mathcal{R} \\
\frac{\Sigma; (\sigma, y) \triangleright B[y/x], \Gamma \vdash C}{\Sigma; \sigma \triangleright \nabla x B, \Gamma \vdash C} \nabla \mathcal{L} \quad \frac{\Sigma; \Gamma \vdash (\sigma, y) \triangleright B[y/x]}{\Sigma; \Gamma \vdash \sigma \triangleright \nabla x B} \nabla \mathcal{R} \\
\frac{\Sigma; B, B, \Gamma \vdash C}{\Sigma; B, \Gamma \vdash C} c\mathcal{L} \quad \frac{\Sigma; \Gamma \vdash C}{\Sigma; B, \Gamma \vdash C} w\mathcal{L} \quad \frac{}{\Sigma; \Gamma \vdash \sigma \triangleright \top} \top \mathcal{R}
\end{array}$$

Fig. 2. The core inference rules of $FO\lambda^\nabla$.

$$B \equiv \nabla x. B, \text{ provided that } x \text{ is not free in } B. \quad (4)$$

We can equivalently state these two axioms as the following inference rules:

$$\begin{array}{c}
\frac{(\vec{x}, b, a, \vec{y}) \triangleright B, \Gamma \vdash C}{(\vec{x}, a, b, \vec{y}) \triangleright B, \Gamma \vdash C} p\mathcal{L} \quad \frac{\Gamma \vdash (\vec{x}, b, a, \vec{y}) \triangleright B}{\Gamma \vdash (\vec{x}, a, b, \vec{y}) \triangleright B} p\mathcal{R} \\
\frac{(\vec{x}, a, \vec{y}) \triangleright B, \Gamma \vdash C}{(\vec{x}\vec{y}) \triangleright B, \Gamma \vdash C} ss\mathcal{L}, a \notin \{\vec{x}, \vec{y}\} \quad \frac{\Gamma \vdash (\vec{x}, a, \vec{y}) \triangleright B}{\Gamma \vdash (\vec{x}\vec{y}) \triangleright B} ss\mathcal{R}, a \notin \{\vec{x}, \vec{y}\} \\
\frac{(\vec{x}\vec{y}) \triangleright B, \Gamma \vdash C}{(\vec{x}, a, \vec{y}) \triangleright B, \Gamma \vdash C} ws\mathcal{L}, a \notin \text{supp}(B) \quad \frac{\Gamma \vdash (\vec{x}\vec{y}) \triangleright B}{\Gamma \vdash (\vec{x}, a, \vec{y}) \triangleright B} ss\mathcal{R}, a \notin \text{supp}(B)
\end{array}$$

Implicit in the above rules is the assumption that variables in local signatures are considered as special constants, much like the nominal constants in LG . The support of B , within a local signature σ , is defined similarly as it is in LG : it is the set $\{a \in \sigma \mid a \text{ occurs in } B\}$.

The logical system with the inference rules in Figure 2 together with $\alpha_{\mathcal{R}}, \alpha_{\mathcal{L}}, p\mathcal{L}, p\mathcal{R}, ss\mathcal{L}, ss\mathcal{R}, ws\mathcal{L}$ and $ws\mathcal{R}$ is referred to as $FO\lambda^{\nabla+}$. In relating LG and $FO\lambda^{\nabla+}$, we map the local signatures to nominal constants, and vice versa. In the following, given a formula B , we assume a particular enumeration of the nominal constants appearing in B based the left-to-right order of their appearance in B .

Lemma 36. *If the sequent $\Sigma; B_1, \dots, B_n \vdash B_0$ is provable in LG then the sequent*

$$\Sigma; \vec{c}_1 \triangleright B_1, \vec{c}_n \triangleright B_n \vdash \vec{c}_0 \triangleright B_0$$

where \vec{c}_i is an enumeration of $\text{supp}(B_i)$, is provable in $FO\lambda^{\nabla+}$.

Proof. Suppose that Π is a proof of $\Sigma; B_1, \dots, B_n \vdash B_0$. We construct a proof Π' of

$$\Sigma; \vec{c}_1 \triangleright B_1, \vec{c}_n \triangleright B_n \vdash \vec{c}_0 \triangleright B_0$$

by induction on $ht(\Pi)$. We consider some interesting cases here:

- Suppose Π ends with id_π :

$$\frac{\pi.B_i = \pi'.B_0}{\Gamma', B_i \vdash B_0} id_\pi$$

The permutations π and π' can be imitated by a series of renaming ($\alpha_{\mathcal{R}}$ and $\alpha_{\mathcal{L}}$ rules). The derivation Π' is therefore constructed by applying a series of α_R , α_L , followed by the id rule.

- Suppose Π ends with $\supset \mathcal{R}$: in this case we suppose that $B_0 = C \supset D$.

$$\frac{\frac{\Pi_1}{B_1, \dots, B_n, C \vdash D}}{B_1, \dots, B_n \vdash C \supset D} \supset \mathcal{R}$$

By induction hypothesis we have a derivation Π_2 of

$$\vec{c}_1 \triangleright B_1, \dots, \vec{c}_n \triangleright B_n, \vec{a} \triangleright C \vdash \vec{b} \triangleright D$$

We first have to weaken the signatures \vec{a} and \vec{b} to \vec{c}_0 before applying the introduction rule for \supset . That is, Π' is the derivation

$$\frac{\frac{\frac{\Pi_2}{\vec{c}_1 \triangleright B_1, \dots, \vec{c}_n \triangleright B_n, \vec{a} \triangleright C \vdash \vec{b} \triangleright D}}{\vec{c}_1 \triangleright B_1, \dots, \vec{c}_n \triangleright B_n, \vec{c}_0 \triangleright C \vdash \vec{c}_0 \triangleright D} *}{\vec{c}_1 \triangleright B_1, \dots, \vec{c}_n \triangleright B_n \vdash \vec{c}_0 \triangleright C \supset D} \supset \mathcal{R}$$

Here the star ‘*’ denotes a series of applications of $ws\mathcal{L}$, $ws\mathcal{R}$, $p\mathcal{L}$ and $p\mathcal{R}$.

- Suppose Π is

$$\frac{\frac{\Pi_1}{B_1, \dots, B_n \vdash C[t/x]}}{B_1, \dots, B_n \vdash \exists x.C} \exists \mathcal{R}$$

It is possible that t contains new constants that are not in the support of C . Suppose \vec{d} is an enumeration of the support of $C[t/x]$. The derivation Π' is constructed as follows

$$\frac{\frac{\frac{\Pi_2}{\vec{c}_1 \triangleright B_1, \dots, \vec{c}_n \triangleright B_n \vdash \vec{d} \triangleright C[t/x]}}{\vec{c}_1 \triangleright B_1, \dots, \vec{c}_n \triangleright B_n \vdash \vec{d} \triangleright \exists x.C} \exists \mathcal{R}}{\vec{c}_1 \triangleright B_1, \dots, \vec{c}_n \triangleright B_n \vdash \vec{c}_0 \triangleright \exists x.C} *$$

where Π_2 is obtained from induction hypothesis applied to Π_1 , and the rule ‘*’ denotes a series of applications of $ss\mathcal{R}$ (for introducing new constants) and $p\mathcal{R}$ (for rearranging the order of the local signature).

- For other cases, the construction of Π' follows the same pattern as in the previous cases, i.e., by induction hypothesis, followed by some rearranging, extension, or weakening of local signatures.

□

Lemma 37. *If the sequent*

$$\Sigma; \vec{c}_1 \triangleright B_1, \vec{c}_n \triangleright B_n \vdash \vec{c}_0 \triangleright B_0$$

is provable in $FO\lambda^{\nabla+}$ then the sequent $\Sigma; B_1, \dots, B_n \vdash B_0$ is provable in LG

Proof. Suppose Π is a derivation of

$$\vec{c}_1 \triangleright B_1, \dots, \vec{c}_n \triangleright B_n \vdash \vec{c}_0 \triangleright B_0$$

We construct a derivation Π' of $B_1, \dots, B_n \vdash B_0$ by induction on $ht(\Pi)$. We show here the interesting cases; the other cases follow immediately from induction hypothesis:

- If Π ends with id , $\top\mathcal{R}$, or $\perp\mathcal{L}$ then Π' ends with the same rule.
- Suppose Π is

$$\frac{\Pi_1 \quad \vec{c}_1 \triangleright B_1, \dots, \vec{c}_n \triangleright B_n \vdash \vec{d} \triangleright B}{\vec{c}_1 \triangleright B_1, \dots, \vec{c}_n \triangleright B_n \vdash \vec{c}_0 \triangleright B_0} \alpha_{\mathcal{L}}$$

By induction hypothesis, there is a derivation Π_2 of $B_1, \dots, B_n \vdash B$. To get Π' apply the procedure in Definition 8 to Π_2 to rename B to B_0 .

- Suppose Π is

$$\frac{\Pi_1 \quad \vec{c}_1 \triangleright B_1, \dots, \vec{c}_n \triangleright B_n \vdash \vec{c}_0 \triangleright C[(h \vec{c}_0)/x]}{\vec{c}_1 \triangleright B_1, \dots, \vec{c}_n \triangleright B_n \vdash \vec{c}_0 \triangleright \forall x.C} \forall\mathcal{R}$$

By induction hypothesis, there is a derivation Π_2 of $B_1, \dots, B_n \vdash C[(h \vec{c}_0)/x]$. Suppose $\{\vec{d}\} = \text{supp}(C)$. Then Π' is

$$\frac{\Pi_2[\lambda\vec{c}_0.h' \vec{d}/h] \quad B_1, \dots, B_n \vdash C[h' \vec{d}/x]}{B_1, \dots, B_n \vdash \forall x.C} \forall\mathcal{R}$$

- If Π ends with $\exists\mathcal{L}$, apply the same construction as in the previous case.

□

Theorem 38. *Let F be a formula which contains no occurrences of nominal constants. Then F is provable in $FO\lambda^\nabla$ extended with the axiom schemes $B \equiv \nabla x.B$ and $\nabla x \nabla y.B x y \supset \nabla y \nabla x.B x y$ if and only if F is provable in LG .*

References

1. J.-Y. Girard. A fixpoint theorem in linear logic. Email to the linear@cs.stanford.edu mailing list, February 1992.
2. L. Hallnäs and P. Schroeder-Heister. A proof-theoretic approach to logic programming. II. Programs as definitions. *Journal of Logic and Computation*, 1(5):635–660, October 1991.
3. R. McDowell and D. Miller. Cut-elimination for a logic with definitions and induction. *Theoretical Computer Science*, 232:91–119, 2000.
4. R. McDowell and D. Miller. Reasoning with higher-order abstract syntax in a logical framework. *ACM Transactions on Computational Logic*, 3(1):80–136, January 2002.
5. D. Miller. A logic programming language with lambda-abstraction, function variables, and simple unification. *Journal of Logic and Computation*, 1(4):497–536, 1991.
6. D. Miller. Unification under a mixed prefix. *Journal of Symbolic Computation*, 14(4):321–358, 1992.
7. D. Miller and C. Palamidessi. Foundational aspects of syntax. In P. Degano, R. Gorrieri, A. Marchetti-Spaccamela, and P. Wegner, editors, *ACM Computing Surveys Symposium on Theoretical Computer Science: A Perspective*, volume 31. ACM, September 1999.
8. D. Miller and A. Tiu. A proof theory for generic judgments. *ACM Trans. on Computational Logic*, 6(4):749–783, Oct. 2005.
9. T. Nipkow. Functional unification of higher-order patterns. In M. Vardi, editor, *Proc. 8th IEEE Symposium on Logic in Computer Science (LICS 1993)*, pages 64–74. IEEE, June 1993.
10. F. Pfenning and C. Elliott. Higher-order abstract syntax. In *Proceedings of the ACM-SIGPLAN Conference on Programming Language Design and Implementation*, pages 199–208. ACM Press, June 1988.
11. A. M. Pitts. Nominal logic, a first order theory of names and binding. *Information and Computation*, 186(2):165–193, 2003.
12. P. Schroeder-Heister. Cut-elimination in logics with definitional reflection. In D. Pearce and H. Wansing, editors, *Nonclassical Logics and Information Processing*, volume 619 of *LNCS*, pages 146–171. Springer, 1992.
13. A. Tiu. *A Logical Framework for Reasoning about Logical Specifications*. PhD thesis, Pennsylvania State University, May 2004.
14. A. Tiu. A logic for reasoning about generic judgments. *Electr. Notes Theor. Comput. Sci.*, 174(5):3–18, 2007.